Bottom-up clustering and top-down shattering of scale-free environments for information fusion

András Lőrincz, Bálint Gábor, Sándor Mandusitz and Zsolt Palotai
Department of Information Systems, Eötvös Loránd University
Pázmány Péter sétány 1/C, Budapest
Hungary, H-1117

Abstract – *We consider information fusion an 'active service', which aims to adapt the presentation of the information to the user. Our work concerns the Internet, a scale-free small world graph, with the tasks being the evaluation of documents, novelty detection and collecting novel documents of 'high value' for the sake of the user. This procedure calls for user-computer interaction. To this end, four algorithms have been designed and are under testing in various Internet environments. The weblog algorithm utilizes competitive value-estimating agents and shatters the Internet domain. Bottom-up clustering develops tree-structured cluster hierarchies and alleviates navigation. Keyword extraction chooses the best keywords that match subsets of the clusters. Link-highlighting makes use of user reinforcement, ranks Internet documents and closes the loop: It provides feedback to the weblog algorithm to improve value estimation and the shattering of the domain. Details about the algorithms are provided.*

Keywords: Scale-free world, information shattering, information fusion, user reinforcement.

1 Introduction

One of the driving forces of information fusion comes from the frustration that we are unable to process, and understand information surrounding us, although some portion of that information may be of extremely high value to us. It then seems reasonable to ask if we could design better software to make computers to *process* the information *for us*. In our view, there are two basic questions to be answered. The first one is about the processing of information. When can we say that information is processed? Assume that we know the answer to this question and the computer makes a perfect job, it 'processes' all available information. Then we have our next question: How much is the value of this processed information for us? Can we use it? In our view, information is processed when it becomes amenable / understandable to human intelligence. In this case, the second question – by definition – is also answered, because *understanding information* means that we can tell its value. This approach aims a higher level of human-computer interaction. The control remains in human hands, because the computers 'goal' is to translate the information for the user. However, the computer becomes an active and interacting partner that analyzes questions, finds out the concepts the user is missing and should 'explain' those somehow.

Another level of interaction arises by noticing that typical information is typically unimportant, whereas novel (not-yet-experienced) information needs to be quickly detected and formulated in terms of known concepts. In most cases, concept development follows a particular route: the observation of novel phenomena and the belonging similarity measures, learning of decision surfaces to categorize the novel phenomena, and – later – a better understanding of the phenomena by establishing their components, i.e., their (early) symptoms. Clearly, learning algorithms and human intelligence need to collaborate here. The solution to this complex problem is not seen at the moment and may require a better understanding of human intelligence. 'Meanwhile', our goal is to develop an adaptive system that works under user reinforcement and fuses information to make it amenable for the user. Our test area is a partially organized database, the Internet.

We are to develop an interaction loop, where concept matching, concept explanation is in the focus to enable human evaluation. Value, on the other hand, may become the driving force of information reorganization. Observing that human associations, the structure of the Internet and most evolutionary systems have scale-free structures, we limit ourselves to the matching of scale-free structures. Associations change from person to person and the matching of the structures may become user dependent.

In what follows, an architecture shall be described that aims to optimize scale-free networks. First, Section 2 explains the basic concepts. Section 3 reviews our basic tools of the matching procedure. Preliminary results are also presented here. The discussion section (Section 4) sketches the architecture that these algorithms shall be embedded into. It is an interacting system that serves the user and works under user reinforcement. Conclusions are drawn in the last section (Section 5).

2 Scale free small worlds (SFSWs)

The last few years have witnessed the evolution of a novel and efficient way of describing complex interactive systems (CIS). The novel description of CIS makes use of graphs with nodes and (directed) edges, representing constituents of the system and the interactions among them, respectively. Classification of CISs is based on the statistical properties of the network. Similar network structures emerge in many different fields. Both these systems and the corresponding dynamical models which define the formation of these networks may be of fundamental importance to understand the behaviors of CISs. The interest in CISs is boosted by the intriguing similarities of biological, social, and information processing networks [1, 2, 3, 4, 5]. The emerging structures show scale-free small world properties.

A graph is a scale-free network if the number of incoming (or outgoing or both) edges follows a power-law distribution ($P(k) \propto k^{-\gamma}$, where k is integer, $P(k)$ denotes the probability that a vertex has k incoming (or outgoing, or both) edges and $\gamma > 0$). A graph is a small world network if (i) its diameter, i.e. the average minimal distance between two nodes, is proportional to $log(N)$, where N is the number of nodes in the graph, and (ii) its clustering coefficient [1] does not converge to 0.0 as N goes to infinity, or alternatively its clustering coefficient is larger with one or two magnitudes than that of the corresponding random network.

Most of the processes, which are considered evolutionary, seem to develop and/or to co-occur in scale-free structures. Importantly, *semantic networks* exhibit scale-free structure, too [6].

3 Algorithms

The *weblog algorithm* utilizes competitive value-estimating agents and shatters the Internet. *Bottom-up clustering* develops tree-structured cluster hierarchies and alleviate navigation. *Keyword extraction* chooses the best keywords that match a subset of the clusters. *Link-highlighting*, which makes use of user reinforcement ranks Internet documents and closes the loop: It provides feedback to the weblog algorithm and enables improved value based reorganization of the documents.

3.1 Finding novel nodes of high clustering coefficients

A particular feature of our task is that novel nodes appear within the scale-free WWW. The direct consequence of the scale-free property is that there are numerous URLs or sets of interlinked URLs, which have a large number of incoming links. Intelligent web crawlers can be easily trapped in the neighborhood of such junctions as it has been shown previously [7, 8]. Our algorithm builds on competing individuals, the *foragers*, in a continuously changing world, where the rate of the emergence of new resources is limited. Fitness of the foragers is not determined by us, fitness is implicit. Similar concepts have been studied in other evolutionary systems, where organisms compete for space and resources and, unlike our foragers, cooperate through direct interaction (see, e.g., [9]). Our foragers are 'intelligent' web crawlers since they crawl by estimating the long-term cumulated profit using reinforcement learning (see, e.g., [10]). The lack of explicit measure of fitness, the lack of our control over the environment, the value estimation executed by the individuals, and the *lack of direct interaction* distinguish our work from most other studies. The algorithm is detailed in Fig. 1(b).

Environment. The domain of our experiments was the world wide web, which is scale-free according to our measurements, too. The distributions of both incoming and outgoing links show a power law distribution. (Fig. 1(a)).

Reward system. Foragers are searching for 'food', which is novel news, and they send them back to the central reinforcing agent, which administers rewards and costs. Positive reward is delivered only to the first sender of a given news item only if the document's time stamp is not older than a day according to GMT. Then reward c_+ is 'provided'. Each sending of a document, costs c_- for the forager. The (immediate) profit is the difference of rewards and costs at any given step (Fig. 1(b)).

Long term cumulated profit. (LTP) Immediate profit is a myopic characterization of an URL. Foragers have an adaptive continuous value estimator and follow the *policy* [10] that maximizes the expected LTP instead of the immediate profit. Policy and profit estimation are interlinked concepts: profit estimation determines the policy, whereas policy influences choices and, in turn, the expected LTP. Here, choices are based on the greedy LTP policy: The forager visits the URL, which belongs to the *frontier* (see below) and has the highest estimated LTP. Visited URLs form a *path* and each path is limited. At each visited page, the forager downloads the neighboring documents and determines whether a document has a time stamp of the current (actual) date. If it does, this document is sent to the center (these documents will be referred as 'sent' documents).

URL lists and decisions. The forager maintains two short-term memory lists, one to avoid loops and another to note the *frontier*, which contains the URL's of pages directly accessible from the visited pages, excluding the visited URLs themselves. A forager has a long-term memory component, the *weblog*, which is a limited list of URLs: those which have proved to be the best starting points of recent paths. At the start of a path the forager makes a random choice from the first 10 elements of the weblog and visits that URL. After a path is finished the forager selects a new starting point for the next path.

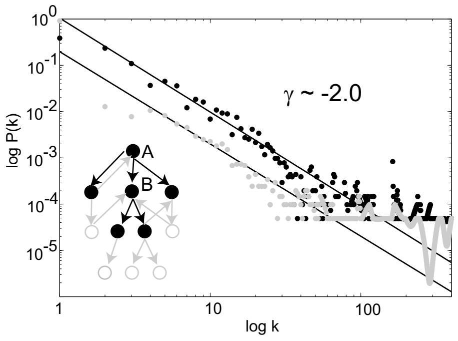
For an URL 'A', the cumulated reward is the sum of immediate rewards collected during the path after visiting URL 'A'. Denoting the cumulated reward of URL 'A' by $R_{path}(A)$, when a path is completed, the *value of the URL 'A'*, denoted by $V(A)$ is estimated as follows:

$$V_{new}(A) = (1 - \beta)V_{old}(A) + \beta R_{path}(A).$$

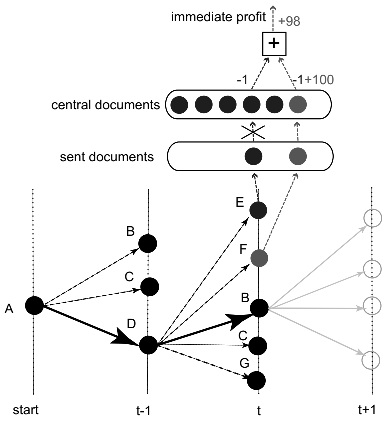
If URL 'A' did not have a value before, then $V_{new}(A)$ is set to $R_{path}(A)$. These values are then used to update the weblog after each path. URLs are ordered by decreasing value and the list is clipped to form the new weblog.

Multiplying by bipartition, extinction and foraging periods. Every forager can multiply by bipartition if it achieves a certain reward collecting rate. The weblog of the parent is randomly separated and passed on to the descendants. On the other hand, if the forager's reward collecting rate falls below threshold then it dies out.

Results are shown in Fig. 2. Time-lag between publishing news and finding those decreases already after a few days (see Fig. 2(a)): the ratio of maxima to minima of the curves increases; and also, fewer news published on Friday were picked up on Saturday, a day late, during the

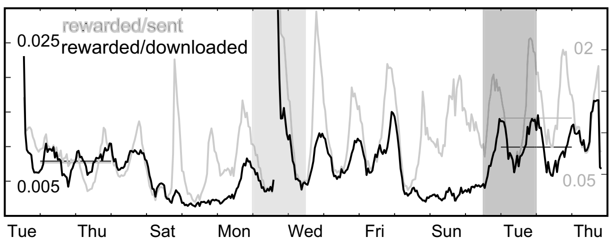


(a) Power law

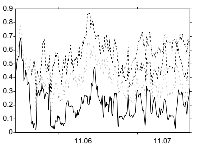


(b) Reinforcement

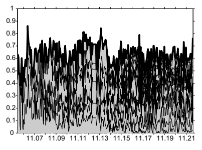
Fig. 1: **Scale-free Internet domain and the rewards**
(A:) Log-log scale distribution of the number of (incoming and outgoing) links of all URLs found during the time course of investigation. Horizontal axis: number of edges ($\log k$). Vertical axis: relative frequency of number of edges at different URLs ($\log P(k)$). Black (gray) dots represent incoming (outgoing) edges of URLs. Slopes of the straight lines -2.0 ± 0.3. *Inset*: method of downloading. The news forager visits URL 'A', downloads the not-yet-visited part of the environment (documents of URLs, which URLs have not been visited yet and are linked at URL 'A'). Downloading is followed by a decision, URL 'B' is visited, downloading starts, and so on. Series 'A', 'B', 'C' $\ldots$ is called path. Black (grey) link: (not) in database. Solid (empty) circle: (not) in database.
(B): Example. Empty (solid) circles: (not) novel documents. Positive (negative) numbers: reward and profit (cost). Vertical dashed lines: consecutive time steps. Dots on the $(t+k)^{th}$ dashed line: documents available at time step $t+k-1$. Path starts at URL 'A'. At start, documents of neighboring URLs 'B', 'C' and 'D' are downloaded. URL 'D' is visited next; this time step is denoted by $t-1$. Documents of URLs 'E', 'F' and 'G' are downloaded. Document of URL 'G' has an obsolete date. Documents of URLs 'E' and 'F' are sent to the center. Document of URL 'F' is novel to the center. Profit 98 is received by the forager. URL 'B' is the most promising and is visited next.



(a) Forager efficiency



(b) Early development



(c) Two week period

Fig. 2: **Experimental results**
(a): The rate of sent and rewarded documents showed daily oscillations. Lighter (darker) gray curve: the ratio of rewarded to sent (rewarded to downloaded) documents. Horizontal lines: average values of the two curves during two workdays. Light (darker) gray regions: number of foragers is 6 (number of foragers increases from 14 to 18). **Division of work: (b,c)** Horizontal axis in 'month.day' units. **(b):** Number of sites visited by only one forager relative to the number of all visited sites in a finite time period ($\approx 75 mins$). Contribution of a single forager is superimposed on cumulated contributions of older foragers. The difference between 1.0 and the upper boundary of the curves corresponds to the ratio of sites visited by more than one forager. Duration: about three days. **(c):** Same for 16 day period. The ratio of different two step trajectories relative to all two step trajectories conditioned that the two step trajectories start from the same site varies between 70-80% (not shown).

second weekend of the experiment than during the first. Further gains in downloading speed are indicated by the relative shift between lighter gray and darker gray peaks. The darker peaks (ratio rewarded/downloaded) keep their maxima at around midnight GMT, lighter peaks (ratio rewarded/sent) shift to earlier times by about 6 hours. The shift is due to changes in the number of sent documents. The minima of this number shifts to around 6:00 P.M. GMT, when it is around 3:00 A.M. in Japan. (Identical dates can be found for a 48 hour period centered around noon GMT.) Maxima of the relevant documents are at around 11:00 P.M. GMT (around 6:00 P.M. EST of the US). During the first week, the horizontal lines (average values of the corresponding curves during 2 workdays) are very close to each other. Both averages increase for the second week. The ratio of sent to reinforced documents increases more. There is a darker grey region in the figure. At the beginning of this region, more crawlers undergo multiplication and the relative shift of the two curves becomes larger. At the end, when multiplication slows down, it is small. Later, because of similar reasons, the shift increases again.

Division of work is illustrated by Fig. 2(b)- 2(c). According to Fig. 2(c) large proportion of the sites are visited exclusively by not more than one forager. Only about 40% of the sites is visited by more than one forager. Fig. 2(b) demonstrates that new foragers occupy their territories quickly. Similar data were found for few (2-4, Fig. 2(b)) and for many (22, Fig. 2(c)) foragers (upper boundary is about the same). The figures depict the contributions of individual foragers: as new foragers start, they quickly find good terrains while the older ones still keep their good territories. The environment changes very quickly, cca. 1200 new URLs were found every day.

The ratio of different two step trajectories relative to all two step trajectories conditioned that the two step trajectories start from the same site varies between 70-80% (not shown). Considering that the first step of the two-steps trajectories are evaluated as 'good' choices by the foragers this ratio shows that foragers exhibit different 'behaviors'. Such differences should be even more pronounced if news of different kinds were searched for. In turn, a relatively large number of sites were visited by different foragers and if sites were visited by more than one forager then they typically followed different paths. Differences between hunting/foraging territories and/or differences between consumed food are called compartmentalization (sometimes called niche formation) The experiments demonstrate that compartmentalization, is fast and efficient in our algorithm.

We note that *clustering coefficients* of elements of the *weblogs* were typically *high.*

3.2 Making order in SFSWs and evaluating the order quickly

Our bottom-up clustering (BUC) algorithm is a local procedure that works by message passages to neighbors, i.e., to nodes that nodes are directly pointing onto. BUC transforms a general scale-free structure into a scale-free tree, amenable for human intelligence. It is a hierarchical agglomerative algorithm based on local decisions. At the beginning, every node is a different cluster. Each of them decides if it is a center of a cluster or if it belongs to one of its neighbors. The decision is based on information requested from the neighbors. In turn, clusters are formed around center nodes. In the next step, clusters are considered as ordinary nodes. Neighbors and their local information are recalculated to maintain scale-free properties. We have investigated the speed of the BUC algorithm by comparing it to our previous top-down clustering (TDC) algorithm. (TDC is a typical top-down graph clustering algorithm which uses global information. In this method the nodes which have the most disjunct set of neighbors are the cluster centers. Other nodes belong to the nearest center node.) Run times of TDC and BUC are compared in Fig. 3. BUC is very fast.

Our *key assumption* is that a clustering algorithm should produce clusters containing documents that are related to each other: A subnet of the Internet can be qualified on the

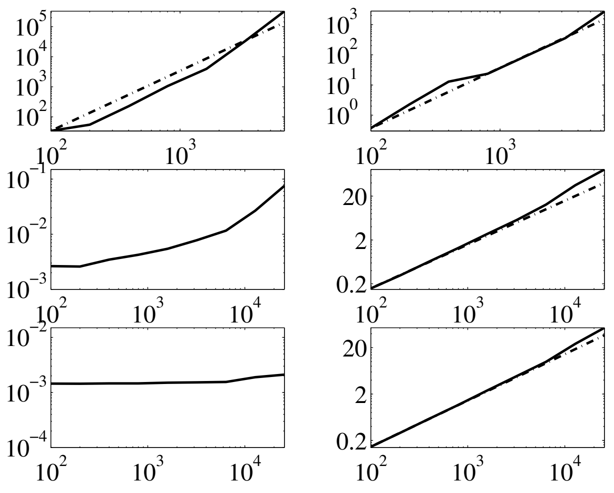


Fig. 3: **Run times of top-down and bottom-up clusterings.**
Comparisons are made on small-world networks generated by the Watts-Strogatz algorithm. Horizontal axis: Number of nodes in network. Vertical axis: Time of clustering in seconds using ordinary PCs Top row: Top-down clustering (TDC). Right (Left): when the best parameters of similarity and size are (not) known. Note that TDC of 3600 nodes took 3 days (left) and cca. 3 hours (right). Middle row: Bottom-up clustering (BUC). Left: estimated running time on parallel PCs. Right: running time on a single PC. Note: The strong non-linear dependence of the left figure is due to nodes formed by BUC that have low clustering coefficients and are connected to most other nodes. This can be avoided by slight modification of the algorithm that keeps scale-free property. Bottom row: Same as middle row but only for the first iteration of clustering. These data provide an estimation of the running time of the algorithm, which keeps scale-free features (see note about middle row).

basis of coherence of documents in the clusters. The evaluation of networks was executed on the following domains[1]:

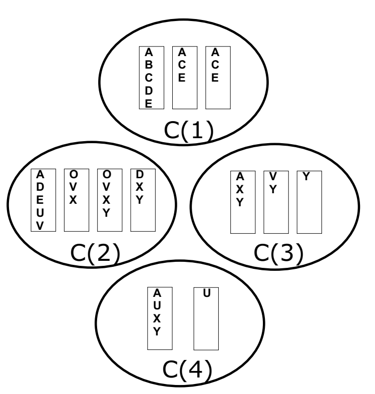
1. American Heart Association (AHA): `http://www.americanheart.org`. It is well equipped with keywords and keyphrases

2. Center for devices and radiological health (General): `http://www.fda.gov/ cdrh/ index.html`. It is moderately well equipped with keywords and keyphrases

3. Collaborative Hypertext of Radiology (Radiology): `http://chorus.rad.mcw.edu`. It is moderately well equipped with keywords and keyphrases

We have used the keywords constructed by the experts of the domain. We used our keyword extraction methods and measured how well the keywords represented their own clusters. Explanation of the clustering figures is provided in Fig 4.

[1]Note that the clustering algorithm may spoil the coherence of documents in the clusters. BUC – under certain conditions – showed similar performance to TDC.



	KW	OR	AND
C(2)	O	1	1
C(1)	BCE	3	3
C(3)		0	0
C(4)	U	1	1
Sum Sum		5	5

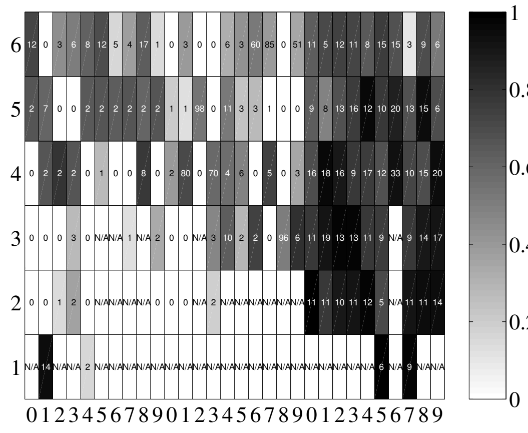
KW =unique keywords of cluster
OR = docs with at least one KW
AND = docs with all KWs

Number of 30% keywords (N3K)	= 5 (B, C, E, O and U)
Number of 'OR documents'	= 3(100%), 2(50%), 0(0%), 2(100%)
'OR' quality	= 100*7/12% = 58%
Number of 'AND documents'	= 1(33%), 2(50%), 0(0%), 2(100%)
'AND' quality	= 100*5/12% = 41%
'Sum Sum'	= N3K and gray level for KW qualities

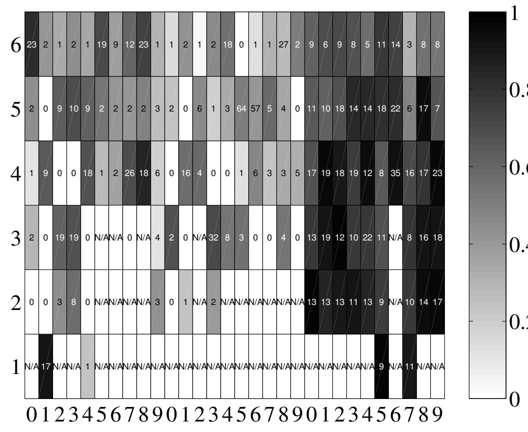
Fig. 4: **Explanatory example for the figures.**
Left: 4 clusters, 12 documents, 10 keywords, *occurrence threshold* (30%) for defining representative keywords, Right: size ordered table with number of keywords in clusters, gray scale coloring showed the relative number of documents containing the below threshold (i.e., good) keywords for 'OR' (i.e., at least one of the keywords) and for 'AND' (all keywords without exception) relations.

Definitions. *Occurrence ratio of a keyword in a cluster*: A document contains keyword K or not. The number of documents of cluster C containing keyword K relative to all documents in that cluster is called the occurrence ratio of keyword K in cluster C. *Keywords of a cluster*: keyword K belongs to cluster C if the occurrence ratio of keyword K is at its maximum in cluster C. In case of a tie, random choice is made between the candidate clusters. The maximum occurrence ratio of keyword K is denoted by $\mathrm{rmax}(K)$. *Threshold*: Threshold is a number between 0 and 1. *Representative keywords*: If maximum occurrence ratio of keyword K is achieved by cluster C, if threshold is set to $0 \leq \theta < 1$ and if the occurrence ratio of keyword K in each cluster $B \neq C$ is lower than or equal to $\theta * \mathrm{rmax}(K)$, then we say that keyword K is representative to the cluster hierarchy at threshold θ. Similarly, if the occurrence ratio of keyword K in a cluster $B \neq C$ is higher than $\theta * \mathrm{rmax}(K)$, then we say that keyword K is not representative to the cluster hierarchy at threshold θ. $100 * \theta\%$ *keyword*: A keyword is called $100 * \theta\%$ keyword, if it is a representative keyword at threshold θ. *'AND' and 'OR' relations of keywords*: If document D of cluster C contains all representative keywords of cluster C then the document is an *'AND'* document. Similarly, if document D of cluster C contains at least one of the representative keywords of cluster C then the document is an 'OR' document. *'AND' and 'OR' qualities of clusters*: The number of *'AND'* documents of a cluster to the number of all documents of that cluster is called the *'AND'* quality of the cluster. Similarly for the 'OR' quality of a cluster. *'AND' and 'OR' qualities of clustering*: The number of *'AND'* documents of all clusters relative to the number of all documents of all clusters is the *'AND'* quality of clustering. Similarly to the 'OR' quality of clustering.

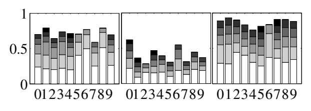
Different versions of the algorithm have been tried. To all methods, 10 different runs were executed to see the variances. We developed methods to balance the size of the clusters (Fig. 5).



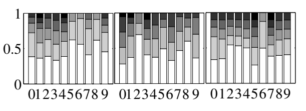
(a)



(b)



(c)



(d)

Fig. 5: **Results of balanced clustering.**
Threshold θ: 0.3. Numbers on the horizontal axis: First (second and third) 0 to 9 columns: AHA, General and Radiology data-bases, respectively. Numbers on the vertical axis: Cluster number. The larger the number, the larger the cluster is. **(a)**: Method 'small clusters do not count' **(b)**: Method 'small clusters join closest relative' **(c)-(d)**: Cluster size distributions in the different runs for databases AHA (left), General (middle) and Radiology (right)

Thresholded (or balanced) clustering algorithm (TCA):

TCA makes use of minimal cluster size parameter s. Given a sequence of clustering phases, that clustering is selected in which there are the most clusters with size larger than s and if more than one such clustering exists, then from these clusterings the one is selected in which there are the least clusters with size smaller than s.

Small clusters do not count: TCA was used first. After TCA has stopped, all small clusters were joined to form a single cluster. Mutual information based keyphrase extraction was applied to the set of clusters formed. Keyphrase evaluation used these keyphrases on the set of clusters not containing the joined one.

Small clusters join closest relative: TCA was used first. After TCA has stopped, small clusters were joined to their 'closest relative'. The closest relative is determined by means of the representative keywords of cluster A - documents of cluster B matrix. The ij^{th} element of this matrix is 1 (0) if the i^{th} keyword of cluster A occurs (does not occur) in the j^{th} document of cluster B. The closest relative of cluster B amongst clusters $A_1, \ldots, A_n$ has the most dense matrix. Closest relative of a small cluster was chosen from the set of large clusters. The small cluster was included into its closest relative. This procedure was followed by keyphrase evaluation.

The balanced methods show improved clustering properties. Representative keywords are present in medium and large clusters, too for all three databases. Further properties of keyword extraction for different Internet domains are shown in Fig. 6

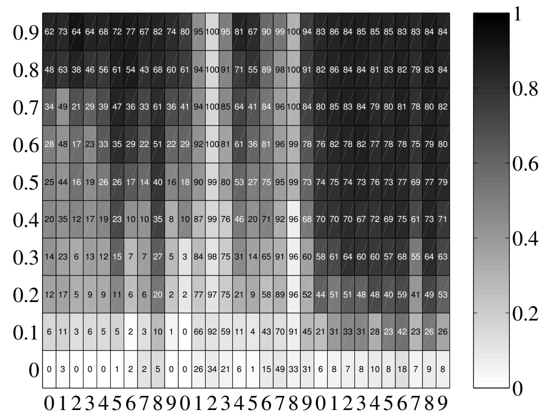


Fig. 6: **All words used for keyword extraction**
First (second and third) 0 to 9 columns: AHA, General and Radiology data-bases, respectively. Numbers on the vertical axis: Threshold values. Numbers in boxes: number of representative key-words chosen from all words. Gray scale: OR qualities of clustering at different thresholds.

Similar results were found for the case when representative key-words were chosen from all keywords (the words of keyphrase metatags) of the databases. As it has been expected on the basis of Fig. 6, our keywords are better on database Radiology than on database AHA. The keywords of database Radiology and the structure of this database are better matched. This could be the result of the highly specified field treated by this Internet domain. AHA is more general and is in between databases Radiology and General. Also, database Radiology is more for specialists, whereas the two other databases are more for the public.

3.3 Representing clusters: Keyword extraction

Keyword and keyphrase extraction algorithms were used for representing clusters and evaluating the efficiency of clustering algorithms. Different methods were tried, including the KEA algorithm [11], words of the 'keyphrase' metatag, mutual information computed for words and mutual information computed for keywords. Computation of mutual information was applied similarly to the technique described by Joachims [12]. It was found that mutual information is about as good or better than KEA with the bottom-up clustering algorithm.

3.4 Adaptation to the user

The last component of our algorithms aims adaptation to user: The algorithm highlights the links in the browser, which most probably match the goal of the user. This way we can help the user by restricting the number of links to choose from. The goal was estimated by the recent steps of the user. The algorithm is based on a set of text classifiers which have output values close to $+1$ or -1. Output $+1$ (-1) means that the input belongs (does not belong) to the text class of the classifier, that is, a classifier can be seen as an expert of its class.

We have assumed that the behavior of any user can be approximated by a relevance (or weight) vector $\mathbf{W}^T = (W_1, \ldots, W_n)$. The estimated value of a document S is the scalar product of the output vector ($\mathbf{S}^T = (S_1, \ldots, S_n)$, concatenated from the outputs of the classifiers) and the weight vector, where superscript T denotes transposition. The goal of the link highlighting system is to optimize its weights to the user. Our assumption allows us – in principle – to adapt to a large number of users. To give a rough estimation, let us restrict the choice of weights to ± 1. Then, for n, the number of possible users is 2^n. Of course, the real issue is not to increase n but to choose the right text classes.

The link highlighting task is to predict the next decision to be made by the user. Two types of questions can be asked during the process. (i) Which is the next document to be chosen by the user? (ii) If we rank the links, how good is this ranking? We shall say that the goodness of our ranking is 90% on average, if on average only 10% of the documents has been given better ranks by the learning algorithm than the document selected by the user. That is, if goodness of ranking is above 90% then about 10% of the documents could be highlighted.

In our experiments the model user corresponded to a classifier. This classifier was excluded from the classifier set used by the link highlighting algorithm. The model user always chose the document which had the highest similarity value according to the single classifier (see later). The link-highlighting system had to mix different classifiers to identify the model user. In some tests the model user could change the topic of interest in order to test the speed of adaptation.

We tested our method with different kinds of classifiers. On a downloaded portion of the Geocities database (i.e., 90,000 html documents) we separated 50 basic clusters by Boley's classification method [13] known to perform well on texts. For classification of documents in the clusters, the probabilistic term frequency inverse document frequency (PrTFIDF) classification method [12] was used. We developed a larger classifier set by building context graphs around the Boley clusters, and training classifiers to its one-step, tow-step, etc. contexts [14] .

Different adaptation algorithms were also tried. Our best method, VEMW technique, utilized moving window and modulated corrections by the error of value estimation. The algorithm is as follows:

$$\delta(t+1) \;=\; \mathbf{W}^T(t)\,(\mathbf{S}^*(t+1)\;-\;\mathbf{S}(t+1)) \qquad (1)$$

$$\mathbf{W}(t{+}1) \;=\; \frac{(1-\alpha\delta(t+1))\mathbf{W}(t)+\alpha\delta(t+1)\mathbf{S}(t+1)}{|(1-\alpha\delta(t+1))\mathbf{W}(t)+\alpha\delta(t+1)\mathbf{S}(t+1)|}, \qquad (2)$$

where α was set to constant to follow online changes, $\mathbf{S}^*(t+1)$ is the best selection according to the value estimation, $\mathbf{S}(t+1)$ is the selection of the user, and $\delta(t+1)$ is the value estimation error computed at surf step $t{+}1$. It is easy to see that the estimated value of the user selected link will be increased by Eq. (2), provided that the $\mathbf{S}$ vectors at each step are normalized $|\mathbf{S}(t)| = 1$ for $t = 1, 2, ...$ In this case, $\mathbf{W}(t+1)^T\mathbf{S}(t+1) \geq \mathbf{W}(t)^T\mathbf{S}(t+1)$.

Performances on the Internet are detailed in Fig. 7

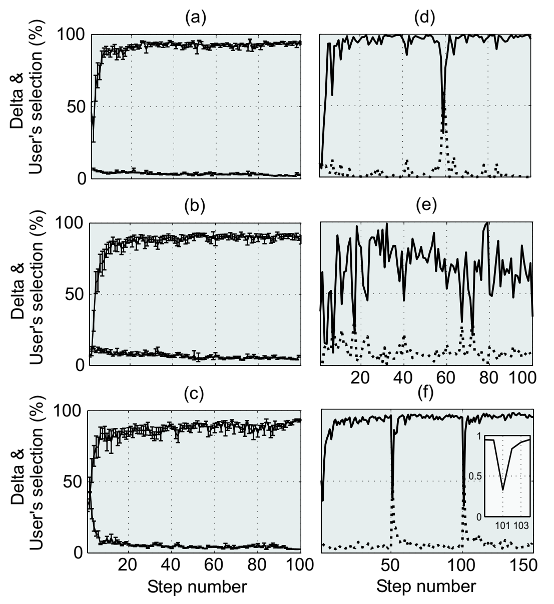


Fig. 7: **Performances and individual cases.**
(a): all Boley classes are used for user estimation, **(b)**: the user's Boley class is excluded from the estimation, and **(c)**: same as b and the set of classifiers used for estimation is extended by classifiers belonging to levels of context graphs. **(d)**: The best highlighting result on the Internet for the case **(b)** of this figure, **(c)**: highlighting with the worst result on the Internet for the same, **(d)**: case **(a)** of this figure with a user changing behaviors. Inset: fast adaptation at step number 100. Upper curves: performances. lower curves: value estimation errors. Error bars denote variances.

4 Discussion

The algorithms that we have introduced or are using, include (i) fast ordering of Internet domains, (ii) keyword extraction, (iii) novelty detection, and (iv) adaptation to and interaction with the user. This interaction can assume different forms. We list a few examples, which could be of use for not-yet-seen domains of the Internet.

1. The user adaptation algorithm, as described in subsection 3.4, develops a weight vector, which contains some information about the user's interest. This weight vector can be used to filter out a certain percentage of the documents. The remaining documents can be reorganized by the fast clustering algorithm and can be presented to the user. Having a set of possible users, the database can be pre-filtered by the appropriate weight vectors.

2. The keyword extraction algorithm can provide keywords for documents, document sets and Internet domains. The BUC algorithm and keyword extraction together can indicate, which portion of the novel domain may contain coherent information.

3. Pre-filtering can be executed by providing keywords, 'and', 'or', 'near', relations amongst them, and so on.

4. Search and pre-filtering of novel documents should be feasible by combining the novelty detecting agent community with one or more of the pre-filtering methods.

These types of interactions are certainly of use for topics, which are well known or at least familiar to the user. They could be of use for topics, which the user is unfamiliar with. However, important information may be filtered out by these methods and care is needed when novel topics are searched for. Novel topics might contain words that we do not understand or have different meanings then those we know. These are typical problems, realized long ago. The worst case scenario is that (i) the material is not having keywords or it is having wrong keywords, (ii) the material is new and we and our 'experts' are unfamiliar with it. In this case, novel methods are in need that have explanatory power and can help in the understanding of the material.

Methods that have explanatory power are rare. One of them is the use of synonyms. Although for novel materials synonyms may not be available in books, the domain that we are looking at, or novel content on the Internet can reveal such relations. Such algorithm has been developed by Turney [15]. The most intriguing approach for understanding unknown phrases of unknown materials has been initiated by Turney and Littman [16]. Their algorithm aims to find analogies to explain the meaning of a word in the given context. In another paper it is mentioned that combination of different modules can be of use for the synonym and analogy problems [17]. In their work, they use statistical information available on the Internet to provide explanations. Clearly, for a fast changing world and for partially ordered knowledge, this should be a useful approach, provided that the new information can be found quickly, e.g., by the fast novelty detecting algorithm.

Novel knowledge should also be matched to the knowledge of the user and the emphasis is to be on the interaction. In our approach, a set of algorithms are made available for the user that can restructure and filter novel information according to the current knowledge of the user.

When the BUC algorithm is applied to novel information, the information is reorganized in a user friendly hierarchical structure and representative keywords are provided for the parts. 'Top-down' modification of the structure is possible by means of keyword-based filtering, adding new links, deleting others, and so on. The BUC algorithm is fast and this process is limited by the user. We call this process as *top-down shattering of information*.

The eventual aim of coupled bottom-up clustering and top-down shattering is (i) to use the knowledge of the user, (ii) to cluster domains and to characterize materials by keyword extraction, (iii) to reorganize materials upon to user reinforcement, e.g., by grading documents, selecting keywords, and so on. Figure 8 depicts the set of our algorithms.

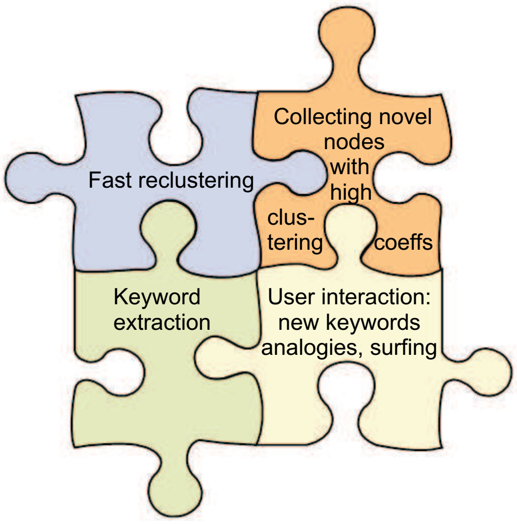

Collecting novel nodes with high clus-tering coeffs
Fast reclustering
Keyword extraction
User interaction: new keywords analogies, surfing

Fig. 8: **Set of algorithms**

5 Conclusion

We have combined four algorithms: (i) fast ordering of Internet domains, (ii) keyword extraction, (iii) novelty detection, and (iv) adaptation to and interaction with the user. These algorithms aim to form a loop for human-computer interaction, which is to optimize user reinforcement and searches, reorganizes for the sake of the user. The individual algorithms have been tested and were efficient. The power of the combination of these algorithms remains to be shown.

Acknowledgements

This work has been supported by the European Office of Aerospace Research and Development (Grant FA8655-03-1-3036). Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the European Office of Aerospace Research and Development, Air Force Office of Scientific Research, Air Force Research Laboratory.

References

[1] D.J. Watts and S.H. Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 393:440–442, 1998.

[2] J. Kleinberg. Authoritative sources in a hyperlinked environment. In *9th ACM-SIAM Symposium on Discrete Algorithms*, pages 668–677, 1998.

[3] A. L. Barabási, R. Albert, and H. Jeong. Scale-free characteristics of random networks: The topology of the world wide web. *Physica A*, 281:69–77, 2000.

[4] V. Latora and M. Marchiori. Efficient behavior of small-world networks. *Physical Review Letters*, 87(19), 2001.

[5] R. Albert and A.L. Barabási. Statistical mechanics of complex networks. *Reviews of Modern Physics*, 74:47–91, 2002.

[6] M. Steyvers and J.B. Tenenbaum. The large-scale structure of semantic networks: statistical analyses and a model of semantic growth. *Cognitive Science*. submitted, http://web.mit.edu/cocosci/josh.html.

[7] I. Kókai and A. Lőrincz. Fast adapting value estimation based hybrid architecture for searching the world-wide web. *Applied Soft Computing*, 2:11–23, 2002.

[8] A. Lőrincz, I. Kókai, and A. Meretei. Intelligent high-performance crawlers used to reveal topic-specific structure of the WWW. *Int. J. Founds. Comp. Sci.*, 13:477–495, 2002.

[9] E. Pachepsky, T. Taylor, and S. Jones. Mutualism promotes diversity and stability in a simple artificial ecosystem. *Artificial Life*, 8(1):5–24, 2002.

[10] R. Sutton and A.G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, 1998.

[11] I.H. Witten, G.W. Paynter, E. Frank, C. Gutwin, and C.G. Nevill-Manning. KEA: Practical automatic keyphrase extraction. In *Proc. DL 99,*, pages 254–256, 1999.

[12] T. Joachims. A probabilistic analysis of the rocchio algorithm with tfidf for text categorization, 1996. http://www-users.cs.umn.edu/ boley/Distribution/PDDP.html.

[13] D.L. Boley. Principal direction division partitioning. *Data Mining and Knowledge Discovery*, 2:325–244, 1998.

[14] M. Diligenti, F. Coetzee, S. Lawrence, C. Lee Giles, and M. Gori. Focused crawling using context graphs. In *26th International Conference on Very Large Databases, VLDB 2000*, pages 527–534, Cairo, Egypt, 10–14 September 2000. http://www.neci.nec.com/ lawrence/ papers/focus-vldb00/focus-vldb00.ps.gz.

[15] P.D. Turney. *Mining the Web for synonyms: PMI-IR versus LSA on TOEFL*, pages 491–502. ECML 2001. Springer-Verlag, 2001.

[16] P.D. Turney and M.L. Littman. Learning analogies and semantic relations. Report ERB-1103, NRC-46488, 2003. National Research Council Canada.

[17] P.D. Turney, M.L. Littman, J. Bigham, and V. Shnayder. Combining independent modules to solve multiple-choice synonym and analogy problems. arXiv:cs.CL/0309035, 2003.