# AN EMERGENT APPROACH TO INFORMATION FILTERING

*M. Mc Elligott, H. Sorensen*

***Abstract***

*With the increasing availability of information in electronic form, it becomes more important to have automatic filter methods so that potential recipients are not flooded with irrelevant details. Internet users, for example, are faced with many sources of information - email, info-servers, news - each with the potential to deliver large volumes of irrelevant data to individual users. In the following paper we present a design for a system to filter news articles on the Internet. The design draws on a wide diversity of theories and techniques in artificial intelligence, cognitive psychology, and artificial life. Among the techniques employed are: dynamic memory architectures, connectionist learning theories, emergent systems, self-organising systems, and genetic algorithms. The work is an attempt to contribute to the rising school of thought in AI at the moment which proposes to unite the fields of symbolism and connectionism so as to produce the structured and flexible design techniques essential for the goal of ever achieving artificial intelligence.*

## 1. INTRODUCTION

The problem with news articles on the Internet is not a new one. Ever since the invention of the written word, the moving picture, the daily newspaper, etc., people have striven to extract from the seemingly endless sea of information around them what is relevant to themselves. Natural filtering methods have evolved in society - newspapers broken up into sections, television programs classed in a certain category and similar Internet users' interests grouped into newsgroups. However it is unlikely that any two users are going to share exactly the same interests, even within a single newsgroup. Consider that in a fairly specific newsgroup, such as that on natural language processing, users' interests can vary quite significantly between semantic analyses, syntactic analysis, pragmatics, etc. - even in different approaches to the same problem. Two people working on the same problem need not necessarily be interested in each other's work. There may be different approaches being employed, different views of what is possible, and so on. The point is that users looking at the cyberspace of newsgroups need their own personal window. They need to see what is relevant to them and avoid what is not. To this end a 'middle-man' (or *agent*) needs to be introduced. There needs to be some process that, armed with a profile of the user's interests, will filter the articles that are posted to the newsgroups (Fig. 1).

The question arises as to what form this filter would take - how could it be designed? Before introducing our own design, let us take a brief look at possible designs that have evolved in the field of general information retrieval. While information filtering may be viewed as a special case of information retrieval [Belkin92], we will describe the techniques as we would see them being employed in the filtering of news articles on the Internet. In reality, they may be being employed for quite different applications within information retrieval [Salton83]. While listing solution strategies individually, it is important to note that, in practice, a number of these may be combined in the search for a general solution.

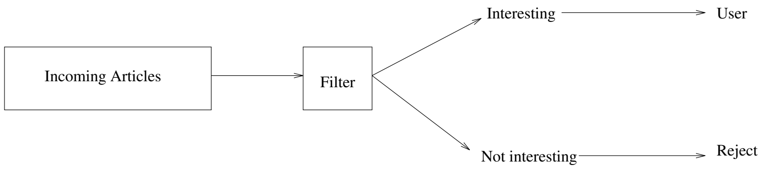


**Fig. 1 - Simple Filter Architecture**

### 1.1 Manual Indexing

This would require a controlled indexing language to be used

> *"in which a single standard phrase or term represents a wide variety of related terms and descriptions. Thus, if the standard entry 'oscillation' is specified in the accepted indexing terminology, alternative related expressions such as 'vibration', 'undulation', 'pulsation', 'swing' and 'rolling' are replaced by 'oscillation' when the documents are indexed" [Salton83].*

So, each article that is about to be posted would need to be reformulated in terms taken from the indexing language. The user can then easily construct their own simple filter which captures their interests by designing it to reject articles that do not contain one or more of these words. Although the approach does offer simplicity of use, it is too labour-intensive and time-consuming to be practical for our needs.

### 1.2 String Searching

Again, a similar idea applies - only accept articles which contain a specific string of characters or words. It is another simple idea which, while not suffering from the problems of manual indexing, does have one big disadvantage. In the words of Blair & Maron, its design

> *"is based on the assumption that it is a simple matter for users to foresee the exact words and phrases that will be used in the documents they will find useful and only in those documents" [Blair85].*

However, Forsyth and Rada point out that

> *"this has never been shown to be possible on large textual databases" due to such things as the presence of synonyms, homonyms, improper syntax and spelling errors [Forsyth86].*

### 1.3 Boolean Information Retrieval

This is slightly better than straightforward string search. A user may design a filter using words combined by the primitives AND, OR and NOT. This scheme inevitably falls prey to the same problem of string searching but remains among the most popular and accurate of the retrieval and filtering methods.

## 1.4 Automated Keywording System

This is a direct attempt to simulate the work of human indexers. It uses a database of key phrases and words. The incoming document is scanned for these and any that are found are passed on to the next phase. This consists of removing ambiguous phrases and inferring from the remaining ones what might be relevant. The remaining list is compared to the user profile list and the article is selected/rejected accordingly. For this design to be able to handle any user, an enormous and impractical database of all possible key phrases and terms that could be employed would be required.

## 1.5 The Vector Space Approach

This approach seems fairly promising and is based on the statistical occurrence of words in both the user profile and the incoming articles [Salton83]. The user profile consists of a collection of words each of which occupies a slot in a *vector* V. Associated with each word is a *weight* which testifies to the extent to which that word characterises the profile as a whole. There is a learning mechanism which concentrates on manipulating these weights. Suppose an interesting article is presented to the system (i.e., it has been tagged as interesting by the user). Previously unencountered words are added to the vector with a low weight value. Words that were already in the vector have their weights increased. Suppose an uninteresting article was presented. Words encountered in this article which are also in the vector have their weights decreased. Finally, suppose it was shown an article and asked to classify it. Words in the vector which also appear in the article are activated and their weights added together to give a value of interestingness for this document.

This technique has many advantages - minimal user intervention, adaptability, robustness, etc. One of the disadvantages commonly associated with it is that articles which the user terms interesting and those which he terms not interesting may not (in fact, most likely will not) be linearly separable. Terms which may be relevant in one context may not be relevant in another.

## 1.6 Expert/Rule Based Systems

This would consist of the user constructing a filtering mechanism out of a set of rules which is then applied to each incoming article. Rules could be such things as:

*(IF the sender is Lenat) and (IF the organisation is MCC) or*

*(IF the organisation is MCB) and (IF the title contains CYC)*

*THEN present this article*

Information retrieval systems such as MAFIA [Lutz90] and the Information Lens system [Malone87] work on something akin to this process and claim good results for restricted in-house applications. The approach does not really have much to offer in the way of a solution to the problem being discussed here. It is not dynamic enough and would require far too much updating to be able to deal with the day-to-day processing of news articles.

## 1.7 Frame Based Systems

In the most likely form this approach would take, a representation system would be required which would consist of an inheritance hierarchy of frames representing events, objects and relationships common to the field(s) in which the user is interested. It would contain inference rules, morphological, syntactic and other information needed for parsing. The user would start off by formulating a description, in natural language text, of their interest(s). This is then passed to the frame network where a succession of frames are activated, each of which contributes to a keyword list. So, we now have a list of keywords which contains the words of the initial query (or description), along with those extra words which were activated in the frame network. This list of keywords is compared against incoming articles to determine which might be of possible interest. Those articles are then further analysed. Each is successively passed to the frame network and the subnetwork of frames activated is compared to the subnetwork of frames activated by the initial user description. The articles are then ranked based on similarity values. The major problem with this whole approach is the tremendous amount of set-up time and maintenance required to implement it.

(note: the above design is a slight reformulation of the ADRENAL Information Retrieval system [Lewis89])

### 1.8 Thesauri

We have come across quite a few mentions of this approach. The underlying idea - that of related words which the user does not specifically use in their description of their interests being added to their profile - is an attractive one and is the basis of all the AI approaches to information filtering and retrieval methods we have come across (e.g. the frame based approach just mentioned). They can prove to be quite useful for domain specific tasks where we could define a finite thesaurus, but they do not really scale up gracefully as the domain to be handled grows.

## 2. PROBLEMS AND PROPOSALS

One of the big flaws with many of the current approaches is that they depend on the user being an expert in describing what is of interest to him/her (which has often proven itself to be harder than it first appears). Although the thesauri-type designs do overcome this to an extent by adding terms related to those that the user entered, unless we possess some thesaurus structure that contains all the knowledge ever learnt about everything and how it links together, there will always be significant gaps in what can and cannot be achieved.

Suppose we did indeed have a very good thesaurus and a user typed in the following as their topic of interest: [ *computer virus* ] - and the thesaurus adds the following terms commonly associated with computer virus:

[ *self-replicating program, malicious code, stealth, boot-sector, partition-sector, infection* ]

Armed with these phrases, the filtering mechanism now separates out articles into those that do contain one or more of these phrases and those that do not. No doubt many of the articles on computer viruses will contain one or more of these terms and be chosen successfully as interesting. The problem, however, is that there may also be plenty of articles which, while not about computer viruses, may contain one or more of these terms. For example, the following phrases could appear in articles:

# "what is the difference between boot-sectors and partition-sectors?"

# "Are cellular automata self-replicating programs? If so, do they constitute
a form of Artificial Life and could they someday acquire a mind of their
own and become malicious towards other life-forms?"

# "I am a biology student and am interested in tracing the infection pattern of
a new virus which has been observed recently...."

etc.

- and be wrongly filtered as interesting to the user.

Our conclusions from this are that the overall semantic content, the proper category an article should be placed in, can only be derived reliably from looking at the article as a whole and not as a collection of scattered keywords. Just because an article contains the term ‘computer virus’ does not necessarily mean the article is about computer viruses; just because an article contains the term ‘object-oriented’ does not necessarily mean the article is about object-oriented programming; and so on.

The only reliable way for a filtering mechanism to decide when the right terms appearing in an article are also appearing in the right context is to have previous experience of just what the right context is. To achieve this end, we intend to design an information filtering mechanism which constructs a user profile based on previous articles that the user has deemed to be interesting. The user profile will evolve with experience - experience of interesting articles.

### 2.1 Episodic Memory

The concept of learning from experience is not a new one - every day we apply lessons we have learnt from previous encounters and situations to handle the problems at hand. One widely held belief in the field of cognitive psychology is that structures exist in the brain to handle stereotypical events. These structures store recurring sequences. For example, a car driving structure would contain the common events that occur each time we drive a car - e.g., 1)open door, 2)sit down, 3)press clutch, 4)put into gear, 5)..... etc. Each event in a structure may itself be another structure; for example, the ‘open door’ event may be a structure containing the events 1)go to door, 2)insert key, 3)lift up handle, 4).... etc. It is proposed that these structures are what guide us in everyday living. They advise us what to do from one moment to the next. Each structure is dynamic and may change as new experiences are encountered. Systems modelled on this design are termed Dynamic Memory Architectures, and they are making an increasingly bigger foray into the field of news article understanding systems. Examples include CYRUS [Kolodner84], SAM [Cullingford78] and SCISOR [Jacobs90].

Much of the credit for the advances and innovations made in this field must surely go to Roger Schank who, along with Harold Abelson in 1978, wrote the influential book "Scripts, Plans, Goals and Understanding" [Schank78] in which they proposed a hierarchical notion of memory in humans. They suggested that, basically, everyday thinking and responses evolved around memory structures known as scripts. These scripts are organised in a hierarchical nature, e.g., menu script inside in restaurant script. The biggest criticism of the idea at the time was its brittleness - scripts could not handle atypical situations which did not conform to any stored script structure. In [Schank82], Schank proposed some additions to the original idea which did not fall prey to the same pitfalls. The following is a brief and simplistic example of the workings of Schank's scripts.

## 2.2 Schank's Scripts

*Step1*:

A novel situation is encountered of which we have no previous experience - take, for example, a parachute jump. The whole experience is recorded in the form of a "live memory" which retains all the occurrences which transpired - what we felt, said, did, etc. Suppose, for simplicity sake, that this live memory contained just five distinct things about the experience and we call these A, B, C, D, E (Fig. 2).

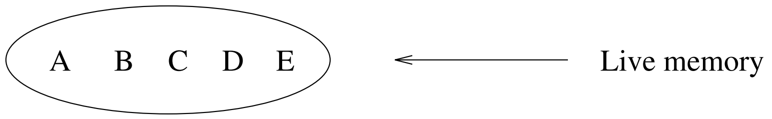


**Fig. 2 - Live Memory**

*Step2*:

This novel situation is encountered again - we do another jump. This is also recorded as a live memory but the process now will be slightly more complicated than before. Suppose this second memory contains A, C, F, G, H (note the variation). We now have the situation outlined in Fig. 3.

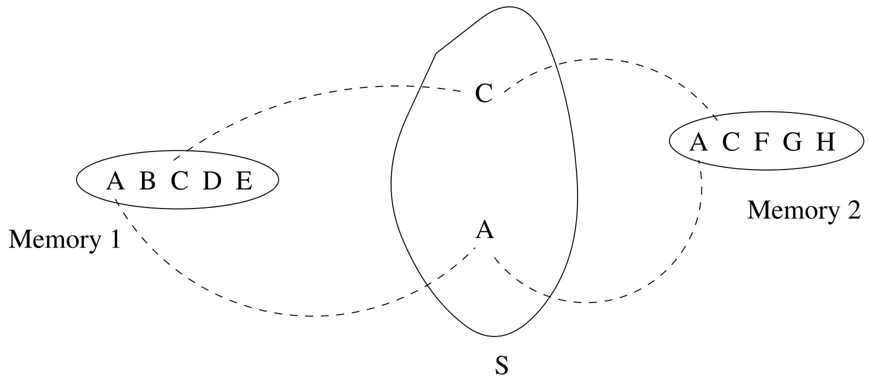


**Fig. 3 - Script with Memories**

Note that a link has formed between similar events in each episode, e.g., A could be 'putting on a parachute'. Schank explained that he could not say how quickly these links (and thus the script) should be allowed to grow and die and that only empirical testing with a computer simulation could give us this insight. In the previous diagram, S is a newborn script and it organises the live memories. We could also look at the first live memory stored as being the newborn script and S as being a more evolved and generalised script. The script will continue to grow and form around recurring concepts. Non-recurring concepts are 'forgotten' (i.e., events which do not 'make it' into the script within a certain amount of time are discarded). Events in the script are gradually strengthened or weakened depending on how much they conform to the regular situation.

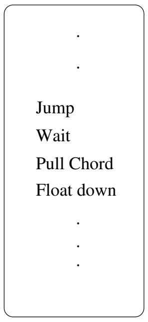


**Fig. 4 - Parachute Jump Script**

Suppose that the script appears as in Fig. 4, and, when we pull the chord, nothing happens. This creates what Schank terms an "expectation failure" and is "indexed" in the script as such. The script may now appear as in Fig. 5.

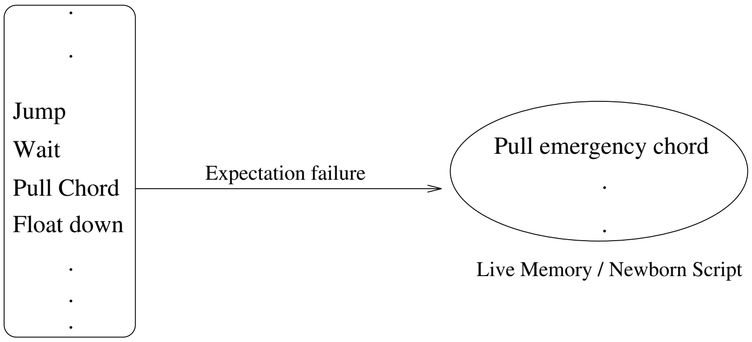


**Fig. 5 - Script with Expectation Failure**

The next time we pull the chord and it does not work, instead of creating another index to an expectation failure, we find that such an index already exists due to a previous experience. This is what occurs when we are reminded of another time. It is this remembered experience which we now use to guide our actions. When we do this we are continually comparing the remembered experience to the present one and it is in this process that a script is formed - similar events in the two situations will form (initially) tentative links. These links are what the script comprises of. If we find that the chord failing becomes a regular occurrence then the original expectation failure will graduate to become an integral part of the mother script (Fig. 6). It is really a case of survival of the fittest. Badly formed expectations are soon discarded for better ones.

.

.

Jump
Wait
Pull first chord
Pull emergency chord

.

.

Float down

.

.

**Fig. 6 - Modified Script**

Now that the whole "pull-emergency-chord" scenario is in the script, we will no longer be reminded of other times when this situation has occurred. This would be something akin to a car-driving script. When you are in the early stage of learning to drive, you are often reminded of your previous couple of lessons - "I remember the last time I was stopped on a hill I had to put on the handbrake", "I remember the last time I was turning off I had to put on my indicator", etc. After a while these responses become an integral part of a script and when you now turn a corner you are no longer reminded of other times you turned a corner, and so on. It is as if we are passengers on a plane with scripts being our pilots and when something abnormal or atypical happens we take over the controls until the scripts are capable once more of handling the situation.

### 2.3 User Profiles - Episodic or Semantic?

Janet Kolodner explains episodic memory as:

> *"The episodic view of memory claims that memory is organised around personal experiences or episodes, and not around abstract semantic categories. In such a memory, repeated or similar sequences of events are recognised as similar and used to build up generalised knowledge about episodes. This generalised knowledge can later be used for understanding. Thus every person's episodic memory will be different from every other person's. Other proposals for memory organisation have stressed "semantic memory" organisations. A semantic memory is a memory for words or concepts that is organised hierarchically using class membership as the major link" [Kolodner84].*

Since, by its very nature, a user's profile - their likes and dislikes - is going to be based on articles they themselves have already read, we believe a good argument exists for designing some sort of dynamic memory system for the formation and storage of a user profile based on the underlying principle of episodic memory. We would argue that most of the current approaches are based on a semantic memory organisation (semantic networks, frames, thesauri, etc.) and do not fully utilise the information that can be obtained from studying and adapting to actual articles that the user finds interesting. The one approach we have come across that does utilise such information is the previously mentioned vector space approach and we intend to make use of this technique combined with others in the new system.

## 3. TOWARDS APPLICATION

Suppose we needed to design a user profile for someone who shows an interest in a specific type of article. We obtain one such article and show it to the profiler. This article is stored as a live memory. Suppose, for simplicity sake, that the article just contains a single sentence comprised of five words - A, B, C, D, E (Fig. 7).

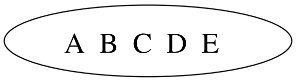


**Fig. 7 - Single-Sentence Article**

[note: the sequence of events here is important. A occurs first, then B, then C, then D and finally E. Another article with the same words but in a different sequence is not judged to be the same article]

This has formed what we will term a newborn script and, for now, this is what the profile will look like. This is the most basic form of episodic memory we can have - a memory built on just a single experience (or, in this case, a single article). The goal of the system is to compare new experiences (i.e. articles) to previous ones and, if they match sufficiently, we take note. Otherwise, unless instructed differently by the user, we ignore it. As it stands, this profile would not be a good reflection of the user's interests - it is simply too narrow and constrained. The profile needs to be enriched by showing the system other interesting articles.

Suppose we show another two simple and interesting articles to the system. The profile now consists of three experiences / articles:

ARTICLE 1 :: A B C D E

ARTICLE 2 :: B F G A D

ARTICLE 3 :: C D F G H

If we suppose that the same words in different articles are somehow attracted to each other, we get a situation which looks something like Fig. 8. We now have a relationship between three articles which the user finds interesting. We have an extremely basic model of an episodic memory system in which frequently encountered aspects of an episode have been recorded in a generalised structure which can be used to guide future processing.

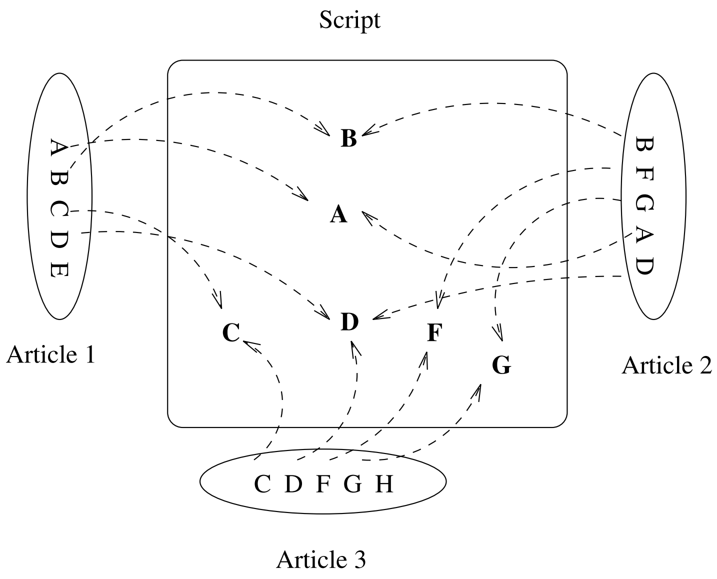


**Fig. 8 - Evolving Script**

There is one important thing missing from this diagram - order! By order, we mean the sequence in which words occur inside the script. For example : A occurs before B (article 1), A occurs after G and before D (article 2). We want these constraints to be reflected in the structure of the script. Consider the following form (Fig. 9).

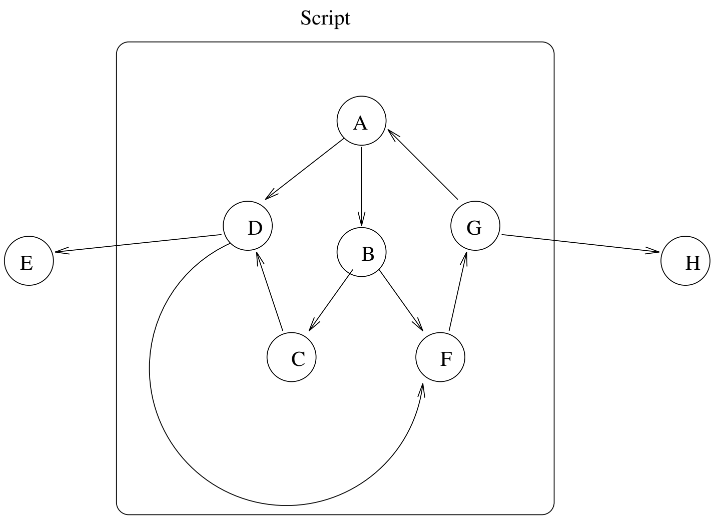


**Fig. 9 - Script with Ordering**

H and E only occurred once in the three articles shown and so remain outside the script structure. If they are reinforced soon by occurring in some other interesting article, then they will begin drifting into the script. If they are not reinforced within a given period of time or articles, they are killed off. As well as being necessary to deal with a changing user profile, this process is an essential and neat way of dealing with misspellings, author names, organisation names, badly structured English, etc.

### 3.1 The Relevance Dilemma

So far, we have discussed how a user profile might possibly be designed based on the script theory of Schank. When we take a closer look at the script that would form from the integration of interesting articles we see one big problem - what use is it? The whole idea behind it was that, when we have this script built, we could compare it to incoming articles and, based on how similar the article is to the script format, we can judge whether or not it would be of interest. When we consider the fact that the system has no idea of what words are - never mind what they mean - we see that there is inevitably going to be words put into the script which are just too general and appear in too many articles to be of any real use in distinguishing articles that are interesting from those that are not. One obvious solution might be to use a stop-list (i.e., a list of words such as prepositions and common terms which occur in a broad range of articles) and to prevent words in this list from gaining entry into the script. There is a problem here, however. Suppose, for example, that we were interested in the latest developments in a specific field. The phrase "state-of-the-art" should distinguish articles about this topic as being particularly interesting but the use of the stop-list would have prevented this phrase from being recognised by not allowing the words "of" and "the" into the script in the first place. Another solution might be to assign a 'contribution value' to each word. Suppose that the total contribution of the words in the script was set at some constant value, e.g. 100. Now, when an article is presented to the system for classification it checks for the presence of the words that are in the script. For the ones that are found to be present in the article they 'activate' and contribute their value. So, for example, the total contribution of the words might come to 63 which, intuitively, is a fairly good outcome and results in the article being classified as interesting. On the surface this may seem like a good solution - words that are in the stop-list can be handled by giving them a negligible contribution value and other words may be given a high contribution value. So, now we could imagine that a person interested in articles on artificial intelligence would have a profile in which the words "Artificial" and "Intelligence" both have high contribution values. Again there is a flaw, and this is best seen from going through a simple example.

Suppose the following article is presented as interesting by the user:

> *"Critics of artificial intelligence propose that the intelligence is not in the system but in the programmer who designed it. They do concede that artificial life is perhaps more ambiguous........... "*

The following are some examples of word combinations that would 'fit into' this:

- ".... of artificial life .."
- ".... artificial intelligence is ...."
- ".... the intelligence propose that ..."

Fig. 10 contains a portion of the profile that will be built from this. Note that the third one (and indeed the first one) may have appeared in articles with nothing to do with artificial intelligence but 'intelligence' (or artificial in the first case) would still have contributed its value to the interestingness of the article since it appeared in one of its allowed sequences (i.e. a sequence that appeared in a relevant article). Clearly, this is undesirable as it opens the door to irrelevant articles being tagged as interesting.

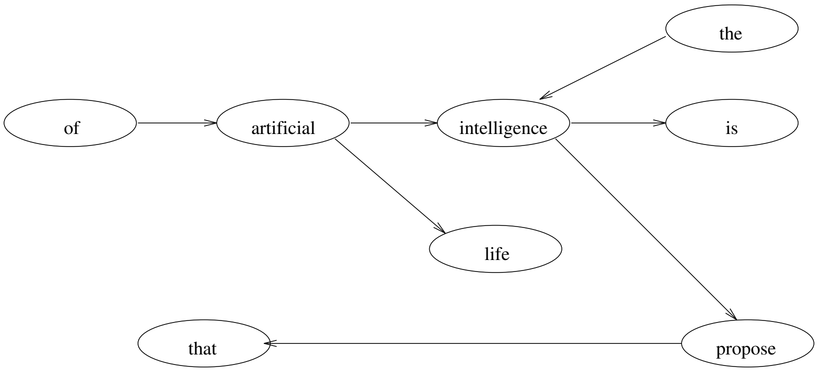


**Fig. 10 - Sample Profile**

We need a system where it is not words that are allowed to contribute, but phrases, and where some phrases are recognised as more relevant than others.

### 3.2 Minds and Minsky

If we disregard scripts for the moment, Fig. 11 is a sort of ‘birds-eye view’ of what a user’s profile will look like.

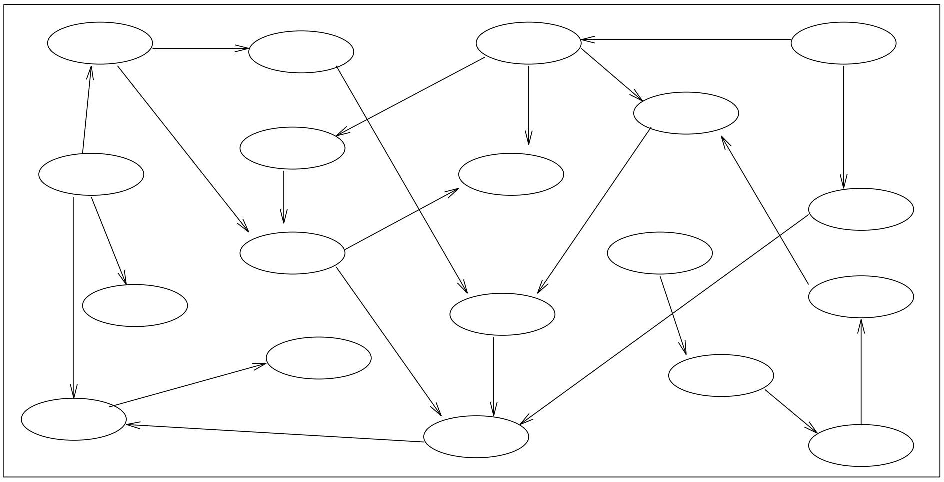

**Fig. 11 - User Profile Structure**

Each node contains a word and the links represent allowable sequences of the words. In "The Society Of Mind" [Minsky87], Marvin Minsky proposed that the mind was a vast society of interacting agents which co-operated and competed with each other to produce intelligent behaviour. This gave us the idea that maybe a document (or in fact the English language, as proposed in [Rowe91] ) could also be viewed as such a society where each word is an agent. Agents co-operate with other agents to form phrases and sentences. To co-ordinate these agents, Minsky proposed a scheme not altogether different from the way events are organised by Schank’s scripts. Minsky put forward a theory of " K-lines " to govern memory and learning. Minsky himself described the theory of K-lines as follows:

*"Whenever you "get a good idea," solve a problem, or have a memorable ex-*

*perience, you activate a K-line to "represent" it. A K-line is a wirelike structure that attaches itself to whichever mental agents are active when you solve a problem or have a good idea.*

*When you activate that K-line later, the agents attached to it are aroused, putting you into a "mental state" much like the one you were in when you solved that problem or got that idea. This should make it relatively easy for you to solve new, similar problems!"*

This gave us an idea on a possible solution to the dilemma of recognising important phrases. Instead of using a single-layer society to represent the user's profile, we use a two-layer one in which this second layer consists of a collection of supervisors 'looking down' on the layer below it. Each supervisor has a specific activation pattern in the first layer to look out for (by doing so, it is really checking for this pattern's presence in the incoming news article) and, when it sees it, then it 'becomes active' and contributes a value to the interestingness of the article. They will form much in the same way that K-lines do. When the society judges a certain sequence of words to be particularly important it forms a supervisor in the second layer to watch out for it. This is sort of like the theory of K-lines in reverse - instead of the K-line activating a "mental state" similar to the one we were in when we got the good idea, it is now the re-occurrence of this previous "mental state" (in the case of this system, the previous good sequence of words) which activates the K-line. Individual words are no longer allowed to contribute to the interestingness rating of the article - only suitably activated supervisors are permitted to do this.

Fig. 12 gives an idea of how the example given in section 3.1 might look in our two-level society. The supervisor depicted remains dormant until the phrase "artificial intelligence" is encountered at which point it 'activates' and contributes its donation to the interestingness of the article. It may be important to note at this point that the supervisor shown is unlikely to be the only one present. There may also be, for example, one waiting for the phrase " artificial life" to appear or perhaps even one waiting for "propose that". Supervisors need not necessarily be waiting for just a two word phrase - there is no physical limit on the size of a phrase but we would not expect one to be formed for a phrase any bigger than, say, five words as it is unlikely that sequences this big are repeatedly going to appear in interesting articles which is the basis on which they are formed.. The remaining question now is just how these supervisors could be formed by the society - how does it decide what sequences are important and what are not?

### 3.3 Deciding on Supervisors

Due to the similarity of the society to connectionist models, we decided to look to the field of neural networks to try and discover some method for the society to be able to pick out important sequences of words. By important, we mean recurring. The phrase "natural language processing" would be considered important (in other words, relevant to the user's interests) only if it appears at a significant rate in articles that the user has deemed interesting. The learning method we have decided on for the society is perhaps one of simplest that neural network researchers have to offer. It is known as anti-*Hebbian* Learning. We will be using an adapted version of this law here and it goes as follows:

*The strength of the link between one word and another depends on how often* *these words occur in sequence in interesting articles. This link is strengthened each time the sequence occurs and possibly weakened when it does not. Each word has a maximum strength which the sum of its link strengths cannot exceed.*

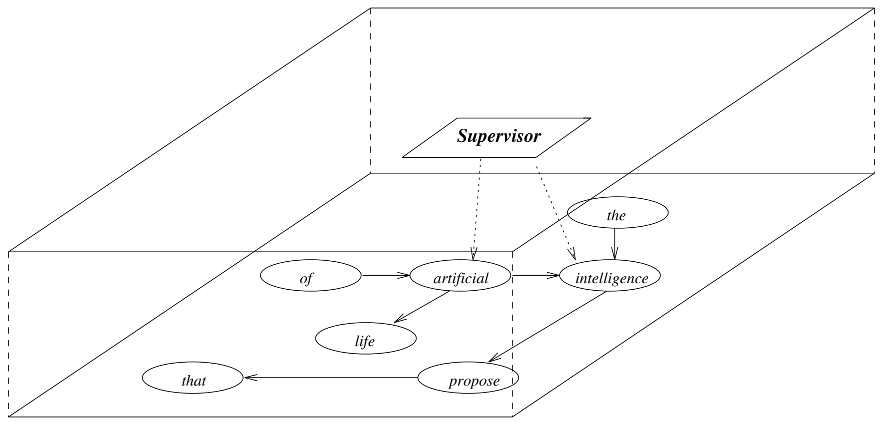


**Fig. 12 - Script with Supervisor**

To see how this strategy will work in practice, we present a simple example. Suppose there was a user who was interested in computer viruses and, in particular, the "stoned virus". They locate an article in comp.virus about the stoned virus and show it to the system. We came across the following article about the stoned virus:

> I recently got the stoned virus on my boot sector. F-protect says it cannot remove the
> virus.

First, the name is F-Prot. Second, which version did you use? Third, there is no such thing as "the stoned virus" - there are dozens of variants and F-Prot usually identifies them quite exactly. What exactly did the program report? Maybe it has been "New or modified variant of Stoned" or "Possibly a new variant of Stoned"? In this case, it is what the program says - a new virus variant (member of the Stoned family of viruses), which is unknown to the program - that's why it refuses to remove it.

> Does anyone know of a way I can remove it without reformatting my hard-drive?
In most cases booting from an uninfected write-protected DOS 5.0 (or higher) system diskette and executing FDISK/MBR will remove a master boot sector infector like the Stoned variants.

Regards,
Vesselin.

Fig. 13 depicts how the word "stoned" would appear in the society. A link between the word and its neighbours is attributed an **X** for each time the link in question occurs in the article.

One way of looking at what has occurred here is to say that a script has formed - the "stoned script". This has formed from experience - experience of seeing "stoned" in an interesting article. Although this script is nothing as complex as those that Schank proposed existed in the human mind, it does contain quite a few similarities. It forms from repeated experience, it contains expectation failures which may graduate into the script (or, indeed, form one of their own) structure if they are reinforced or die off (forgotten) if they are not and, finally, these scripts organise word patterns much in the same way that Schank's scripts organise events. It is the formation of these scripts that determine where and how supervisors are created. They are there to look out for script formations, each one being monitored for by a specific supervisor.

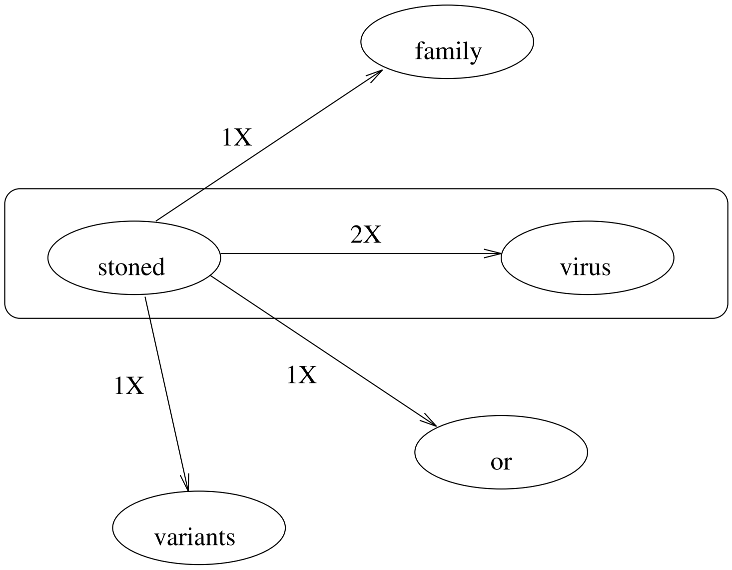


**Fig. 13 - Weighted Script**

## 4. PUTTING IT ALL TOGETHER - ORDER FROM CHAOS

This may all appear a bit haphazard. You may be thinking that the recognition of important phrases cannot be accomplished by mere frequency alone - what about the phrase "to the"? This will undoubtedly have a high frequency in any article that is presented; so a supervisor will be created to monitor for it. Here we have a problem. This phrase is going to appear in a wide variety of articles and not just those on the stoned virus, so how do we make sure that when this supervisor activates it only makes a negligible contribution to the interestingness of the document? The trick is to have variable contributions. What we would like is for those sequences of words that help to distinguish the article (as interesting or not) the most accurately to make correspondingly bigger contributions when they are encountered. For example, we would like the supervisor for "stoned virus" to make a bigger contribution than the one for "to the" when they are activated. This could be accomplished as follows: Suppose that our system has built a society model based on some articles that the user has presented as interesting. Now the system monitors the articles being posted to newsgroups and picks out those that have activated a collection of supervisors such that the resulting interestingness value is above a pre-defined threshold. These articles are presented to the user who rates them as relevant or irrelevant. For those articles that were rated as relevant, the system checks what supervisors were activated by this article and thus contributed to the interestingness of it. These supervisors are rewarded, i.e. their allotted contribution values are increased and those of the other (unactivated) supervisors are proportionately decreased. In the case of those articles that were rated as irrelevant by the user, a similar operation is carried out - those supervisors that contributed to it have their allotted contribution values decreased while the others have theirs increased. The result of this will be that the best sequences of words (best in that they capture the user's interest) make more of a contribution to the interestingness of the articles.

## 5. ADAPTING AND EVOLVING

The success of the design largely hinges on the quality of the articles that the user presents to teach the system. If good articles are presented which capture the user's interests well, then the design should, as a consequence, produce good results. If, however, the articles are not good enough then the results will suffer. This could pose a significant problem - how does the user know what is good enough? Returning to the stoned virus scenario, imagine that there were two users (each using separate implementations of the system) - one interested in computer viruses in general and another interested in just the stoned virus. They may both present the same article to the system as interesting since they are both interested in the stoned virus. While this may be fine for the first user, it may well prove to be too specific for the second one. You might say that all the second user needs to do is to present a few articles about different viruses and so build up a profile resulting in any articles on computer viruses being picked as interesting. This presents yet another dilemma - what about those users that may be interested in just a specific group of viruses and not all of them. If the system generalises too much from articles presented by this user it may never be able to pick out a certain category as interesting. This problem, and ones like it, will continue to surface right across the board. What we need is for each implementation to customise itself to each user. For example, one implementation may have more agents in order to capture a wider field; another might have a quick link formation in order to capture a specific interest straight away; another might have a slow link formation to allow it to generalise sufficiently from the presented articles; and so on. All in all, we envision somewhere between five and ten different variables that could come into play, e.g.

# MAXWORDS : maximum number of agents in the society;

# MAXLINKS : maximum number of associates an agent may have;

# STRENGTH_ALLOC: amount the occurrence of each link is worth;

# THRESHOLD : lowest strength which link is allowed to survive at;

etc.

What we would like is for each implementation of the system to optimise these settings for its respective user. The solution, in our opinion, lies in the field of Artificial Life i.e. mimicking the activities of biological life-forms using computer simulation. To see how we may apply ideas from this field to the application at hand, consider the following scenario which has long been suggested as the most plausible theory of biological evolution (Fig. 14).

- an organism is going to be surrounded in its environment by both life-enhancing and life-threatening stimuli. Only the fittest (those with the higher life expectancy) survive, i.e. those that have adapted themselves better than the others to get more enhancing and less threatening stimuli. These surviving organisms then mate, reproduce, and thus pass on their adapted genetic code to their offspring which are part of a new generation of competition and survival.

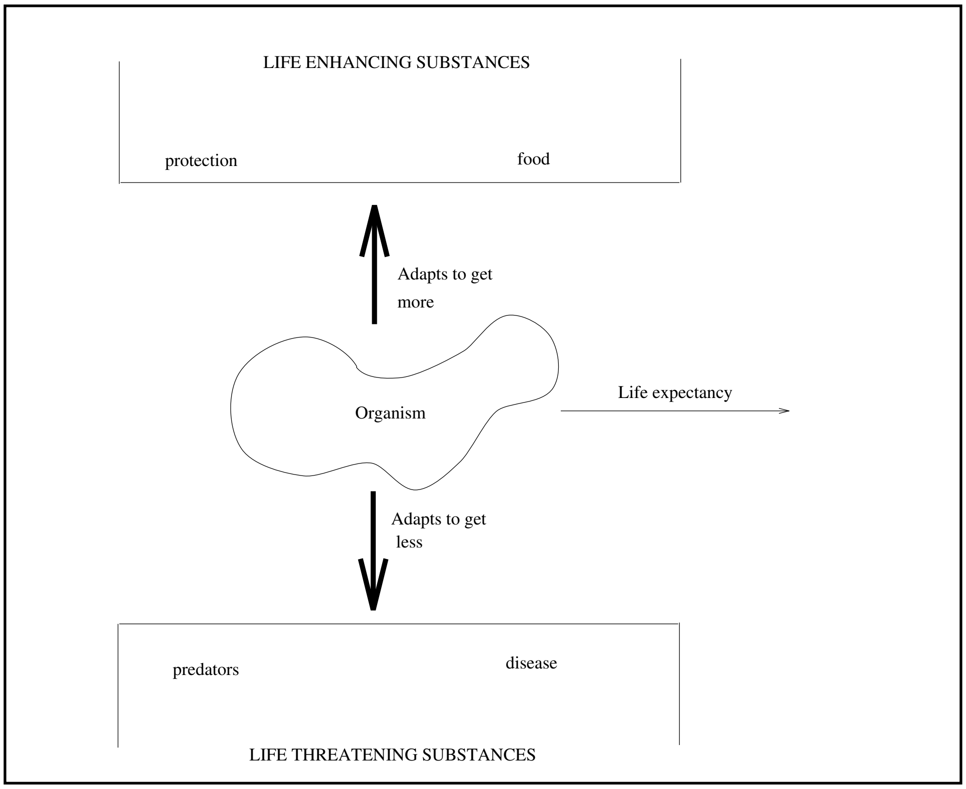


**Fig. 14 - Evolutionary Adaptation**

We can transplant this idea almost directly to deal with the problem at hand. We could say that there is a direct correlation between:

(a) the organism and the society model;

(b) the life-threatening substances and uninteresting articles;

(c) the life-enhancing substances and interesting articles;

(d) the life expectancy and the quality of results of the system.

The question now is, how do we get society models to compete, mate and reproduce with the overall goal of the best possible model evolving? The solution lies in the field of genetic algorithms.

### 5.1 The Process of Evolution

If we consider the society variables mentioned on the previous page, we could imagine the model having a sort of primitive genetic code comprising of between five and ten types of genes. To illustrate this feature, we use a hypothetical example :

A user sits down for the first time with the system and presents, say, three interesting articles to it. The system creates a number of society models based on these articles. Each model is built using randomly selected values for their variables - each has a randomly selected genetic code. Each model then scans the news server for, say, the next twenty incoming articles and rates each one as interesting or not. Each model in turn compares its choices to the user's choices and calculates a score (based on the success rate). The genetic algorithm now comes into play and selects those with the highest scores (fitness values) to mate and reproduce resulting in the birth of a new generation of society models whose average fitness will be higher than that of the previous one. Each model in this new generation is now told to scan those same articles that were judged by their predecessors. Again, each calculates its score. This process of evolution continues until the creation of a model which has an acceptably high fitness value (e.g. has rated seventy percent of the articles correctly).

It may happen that, after repeated tries by the genetic algorithm, it finds that it cannot achieve the desired success rate. In this case, we inform the user that the articles originally presented to the system were not sufficient to build an adequate profile and request them to provide the system with more information (i.e. interesting article(s)). Methods are being researched at the moment which may help the genetic algorithm decide which variable was degrading the results and if this is possible then we will be able to present the user with more specific problem evaluations such as - articles too general; articles too specific; articles too small; etc.

## 6. FINAL COMMENTS

A design for an information filtering system has been presented which promises significant advantages over current approaches, for example: it is compact compared to other methods designed utilising AI techniques (since it does not use any large thesaurus-like structure to try and derive the semantics of an article), it is easy to use since the user needs know nothing of how the system works - they merely need to present interesting articles and, finally, it gives us an insight into how a hybrid system based on symbolist and connectionist theories along with AL techniques might be designed to provide a structured, flexible, and adaptive synthesis.

**REFERENCES**

[Belkin92] Belkin, N.J., Croft, W.B. "Information Filtering and Retrieval: Two Sides of the Same Coin",Communications of the ACM, Vol. 35, No. 12, December 1992.

[Blair85] Blair, D. C. and Maron, M. E. , "An evaluation of retrieval effectiveness for a full document retrieval system," Communications of the ACM, Vol. 28, No. 3, March 1985.

[Cullingford78] Cullingford, "Script application: computer understanding," Technical report, Yale University, 1978.

[Forsyth86] Forsyth, R. and Rada, R. , "Machine learning: Applications in Expert systems and Information Retrieval," Ellis Horward London, 1986.

[Jacobs90] Jacobs, P. and Rau, L. , "Scisor : Extracting Information from On-Line News," Communica tions of the ACM, New York, Vol. 33, No. 11, November 1990, pp. 88-97.

[Kolodner84] Kolodner, J., "Retrieval and Organisational Strategies in Conceptual Memory: a Computer Model," Erlbaum Hillsdale, N. J. , 1984.

[Lewis89] Lewis, D. D. , Croft, W. B. and Bhandaru, N. , "Language oriented information retrieval," International Journal of Intelligent Systems, Vol. 4, pp. 285-318, 1989.

[Lutz90] Lutz, E. , Kleist-Retzow, H. V. and Hoernig, K. , "MAFIA - An Active MAil-FIlter-Agent for an Intelligent Document Support," Multi-user Interfaces and Applications, Elsevier Science Publishers B. V. (North Holland), 1990.

[Malone87] Malone, T. W. , Grant, K. R. , Turbak, F. A. , Brobst, S. A. , Cohen, M. D. , "Intelligent Information-Sharing Systems," Communications of the ACM, Vol. 30, No. 5, May 1987.

[Minsky87] Minsky, M., "The Society Of Mind.", Simon & Schuster, 1987.

[Rowe91] Rowe, J. and Mc Kevitt, P. , "An Emergent Computation Approach to Natural Language Processing," Irish Conference on Artificial Intelligence and Cognitive Science, University College Cork, Ireland, 19-20 September 1991.

[Salton83] Salton, G., McGill, M.J., "Introduction to Modern Information Retrieval", McGraw-Hill, 1983.

[Salton86] Salton, G., "Another Look at Automatic Text Retrieval Systems," Communications of the ACM, Vol. 29, No. 7, July 1986.

[Schank77] Schank, R.C., Abelson, H., "Scripts, Plans, Goals and Understanding," 1977.

[Schank82] Schank, R.C. "Dynamic Memory: A theory of reminding and learning in computers and people," Cambridge U. P. London, 1982.