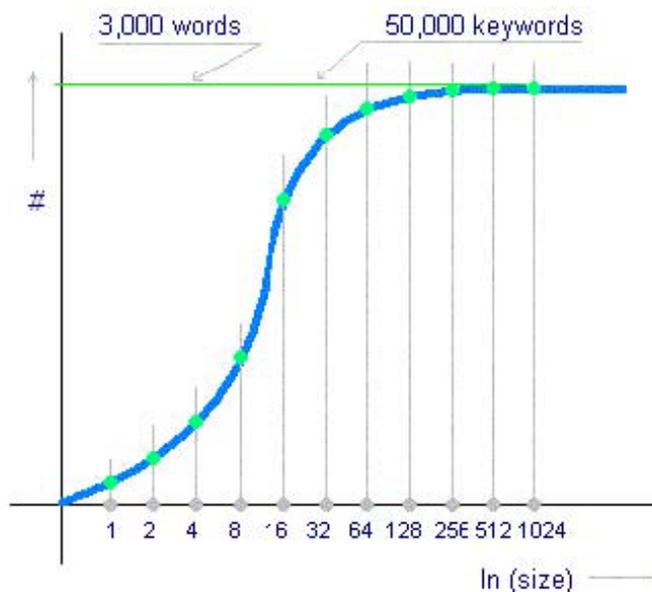


## How to build Human Knowledge Maps

### Algebraic Type Operators you are going to need

Juan Chamero, [juan.chamero@intag.org](mailto:juan.chamero@intag.org), as of October 20th 2004-10-16; see [Prototype](#)  
It was reviewed, as of April 2006.



$P\{r;8\{G\{1,000;C\{K500\{F1\{(\text{Google},1,000);E(T),HK\}K)\}}\}}\}$

Graphic outcome of a Parsing Operator (P) process that unveil "Common Words" (3,000) and keywords (53,000) from of a Library of Documents belonging to a single discipline of a [HKM](#), Human Knowledge Map. Along the x coordinate of this peculiar "S curve" we depict the logarithm of documents' sample size times a Unit Sample Size versus the number (#) of detected semantic units, being either Common Words or keywords. . .

## Introduction

The process we are going to describe is the core of Darwin<sup>®</sup>, a Knowledge management methodology that belongs to [Intag, Intelligent Agents Internet Corp](#). Darwin stands for Distributed Agents to Retrieve the Web Intelligence, and it is a multi-agent platform that operates via an e-membrane<sup>®</sup> built in an Expert System named FIRST, Full Information Retrieval System Thesaurus<sup>®</sup> both also belonging to Intag. Darwin's algorithms and algebraic operators described hereinto work over Sparse High Order Boolean Matrices in a semantic three dimensional space: documents, subjects, and keywords of sizes 10 billion, 500,000, and 10 million respectively.

Let's see the Realm where these operators will perform their mission. One world wide realm will be the Web space. In the Web space billion of documents are indexed and referred in some special Websites known as "Search Engines". The Web space is far from being considered a Library because documents are hosted in sites at will, without any type of permission and control. The Web is absolutely free. Of course Websites are more or less organized, more or less specialized and with different levels of popularity and traffic. Knowledge is disperse all over the Web, no "mapped" at all. Concerning a specific discipline (Major Subject of the Human Knowledge), for instance medicine, we may find via Search Engines more than 40 million of documents. Why so much?. Because any of those referred documents have the "word" medicine cited one or more times within their content.

From time to time appears in the Web Community some knowledge organizers known as “Virtual Libraries” that try to make a meaningful Cognitive Offer of main authorities of a given discipline. Most of them are voluntary efforts that accept the not always bona fide Websites owner’s promotion of their respective contents. These Virtual Libraries generally offer only a few thousands of references. To provide more and better information to robots that continuously crawl the Web space gathering references for their respective Search Engines some special command were added to HTML like Meta Tags, where Web page owners are allowed to include some crucial “keywords” supposedly present in the document as an aid to guide users’ preferences and needs. However, also these bona fide features proved to be useless, mainly because Website Webmasters’ abusive and deceiving bad use to mislead people.

As a better approach some Search Engines proceed to browse by themselves document by document parsing them as analytically as possible, from their beginning to their end. In this way the indexing task is more objective. They do even more in order to provide people better information about sites: their “popularity”, a measure of their relative importance in the Web space. As Internet is democratic Community they alleged that “voting democratically” among Websites would be the best possible way to measure their relative authoritativeness. A vote is a mention of a given site. It is reasonable that sites that are most referenced by the rest of the Community could be considered the most authoritative even though restricted to this type of popularity metric.

### ***The Keyword concept arises soon as a crucial need in Information Retrieval***

Another bettering came through the advent of keywords. Keywords are a single word or a precise string of words that have precise and consensual meaning. If our languages were ideographic keywords would be represented by ideograms. Keywords evolve too much along the time. It’s estimated that there are nearly 10 million keywords in a language like English, of course by far much more than “Common Words”. These keywords are present in the Web space, some ones in a few documents and some ones in millions. ***However documents hosted in the Web usually are not indexed by keywords in conventional Search Engines but by words!. That’s a remarkable difference.*** Of course if you query for a “modal” and specific keyword (remember meaningful and consensual) you will succeed obtaining a packet of good references!. Most Search Engines offer something like a “guess your keywords” feature enabling you threading logical chains, for instance by enclosing the words string that describes your guessed keyword between quotation marks. If you query by parallel processing the Search Engine will probably bring you documents indexed by “parallel” and by “processing” but if you query by “parallel processing” instead the Search Engine will probably select those that have within their content that exact string (with certain flexibility that changes from a Search Engine to others).

### ***Keywords are “adopted” by Authorities***

We would like to emphasize the word “you” above because the selection of right keywords rests on you, the people. Most conventional Search Engines knows almost nothing about real keywords (some Search Engines and some related “Word Tracking” services are now accumulating experience keeping statistics about the most frequents “words” used by people). To make things worse, as we have commented above, documents are not indexed by keywords yet, fundamentally because they do not know how to discriminate them!. If you as a user fail to guess the right keyword you may fall in a black hole of uncertainty, noise and ambiguity.

In summary Search Engines industry faces two challenges, how to offer an objective cognitive offer and how to guide people to get what they need efficiently, let’s say in no more than three clicks. The popularity algorithms provide an approach to authoritativeness as long as the effect of perverse marketing is minimized. Frequently “truth” and “values” with poor marketing are hidden, kept too much from users’ awareness because their relative weights were too low. For instance, a good article originated in a Finland Scientific Journal has to be replicated in an American Scientific Journal to be known (to be high ranked) because a significant difference in their respective popularities. The popularity metric measured in this way is too sensitive to marketing campaigns.

At the same time and as a coupled joint effect authorities that rank high are more authoritative to foster new keywords (“modal” keywords) and other authorities under their area of influence tend to adopt, use and strongly recommend their use because their authoritativeness and because marketing influences as well. A dangerous positive feedback process tends to show more and more as visible truth the ranked high references. Let’s suppose that you look for a scientific paper about a very specific subject via your guessed keywords. It may happens that almost all 10 Top references brought by Search Engines be useless!. What happened then?. Irrational and abusive keywords marketing is distorting and in some extent contributing to collapse the Web.

### ***The Subject concept – a new semantic dimension***

Up to now we were talking about words and keywords. We have notwithstanding a superior cognitive category: the subject. It is a concept that has the status of a piece ok know-how, a prescription, sometimes a procedure, a special talent, in fact something that in a given culture has acquired the status of something valuable to learn and that is formally taught by teaching authorities. In this sense subjects are items of disciplines Curricula, programs, Syllabus, and menus of activities. Subjects are usually structured as Logical Trees. Of course a subject is by “de facto” a keyword but the reverse is not always true. Only some keywords are or became subjects along knowledge evolution. Disciplines and their subjects are present in the Web but they are not explicitly shown.

### ***How a single discipline is present in the Web***

It is estimated that 200 disciplines are present in the Web belonging to 9 branches of the Classic Human Knowledge. Disciplines are formally represented by Logical Trees (Ontological) where we may identify the following components:

- The root
- The branches frequently split in 4 to 6 levels with their corresponding nodes
- Leaves, as terminal nodes
- Names of all those components
- Representative samples of authoritative documents in each node

### ***Example***

In figures what we are talking about

Discipline – Major Subject: for instance Physics, Medicine, Philosophy.

Subjects: From 800 to 1,600, 1,000 in the average among branches and leaves

Levels: frequently from 4 to 6

Sample of documents (authorities), written in a given Jargon, representing the discipline cognitive content: from 200,000 to 500,000 documents with a minimum level of authoritativeness.

A Starting Sample (the first step of an exponential binary content analysis): 10,000 documents

A Binary Progression Sampling: for instance 1, 2, 4, 8, 16, 32 times the Starting Sample, which leads to a Jargon Sample of 320,000 documents

Each document “weighting” in the average 2,000 words

Words: 640,000,000

Documents per subject: 320, whether existent. As we go deep down the discipline tree documents tend to diminish rather “exponentially”.

Estimated keywords: 50,000

Jargon: Common Words set used to document the discipline documents: ~2,000

Documents inner structure: They are strings of the following type:

[w\*w\*w \*w\*w\*k\*w\*w w \*i\*w\*w\*k\* .....],

Where

w, stands for word, single word

k stands for keyword (defined by a string of one or more words, unknown, see note 1)

i stands for image, picture, any multimedia object

\* stands for separator or punctuation mark, end of line, end of paragraph,...

Note 1: keywords are initially unknown. In this analysis images will be ignored.

In the average we may estimate (n) n-ad chains of "Common Words" per "suspected keyword", being (n) the maximum words size of the suspected (see Note 2).

N-ads: 2560,000,000 for n=4

Expected different keyword citations per document: from 10 to 30

Average keyword repetition within a document: 5

Average appearance of the most frequent keyword: 1,600, it's supposed that at least one keyword is common to all documents within the same subject, so it would be present 320x5 times in its sample library niche.

Keywords per subject: 50

Average appearance of a common word in the subject sample:  $640,000,000/2,000 = 320,000$

Potential keywords are by far more infrequent than common words in 1:200 ratio. It means that the relative appearance of keywords is extremely low even in big samples:

Total amount of n-ads: 2560,000,000 (for n = 4)

A problem arises here: how many n-ads (monads, dyads, triads, ..., n-ads) are different, distinguishable?. Within them should be the 50,000 keywords "flat Thesaurus"! How to discriminate them?, how to distinguish them from "rare Common Words"?. Rare words occur with a very low probability but evenly distributed along the whole discipline. On the contrary, keywords are very specific; they generally "cluster" around a very specific subject, rarely around two or more.

Note 2: n-ads: considering paragraphs strings of p words in the average, the amount of n-ads is given by the expression:

$$\#(n\text{-ads}) = np - n(n-1)/2$$

Where #(n-ads) is bounded by np. We may imagine then any document (without images, and separation symbols) as a single string of p words, for instance 2,000 words and then bounded by  $2,000n$  n-ads.

## ***The process of "killing" Common Words***

Given as known the CW[cw] Common Words list we may proceed to "kill" them (see note 3) document per document leaving strings of the following type:

[ \$ \$ w \$ w \$ \$ \$ w w w .....]

Where symbol \$ stands for a killed common word. So w\$w could be two separate single keywords and/or a triple word keyword replacing \$ by the right "connector", for instance "of" rendering the keyword "w of w".

Note 3: In CW[cw] those words that could form part of a keyword should be marked in order to avoid killing them in a first scan checking that are not part of a keyword. For instance the word "parallel" could be considered a common word if not followed by "process" that is another common word.

## **Algebraic Operators Draft**

We are going to define a set of Algebraic Operators to be used by agents that handle complex data strings and sets. Agents' Tasks could be very sophisticated (see Appendix: Agents' Tasks).

### **Fetch References Operator**

Go and Get certain data is its mission. The "go" portion points to a special Data Reservoir, for instance a Search Engine Database. To accomplish this complex task the agent should know precisely how the reservoir is structured in order to arrange its "bait and catch" data retrieval operation.

**F (settings; arguments):** F applies over some reservoir and consequently F must be settled to communicate with it via a set of parameters. Arguments stand for "what to fetch", the final object of fetching. Argument could range from a given keyword to a string of logical operators (even fuzzy ones), depending of how the reservoir is structured and its purpose, the goal that guided its creation.

**F1 ((SE, n); k)** would be an example of a particular fetch, let's say a type of fetching, namely F1: applying over search engine SE, for instance Google, to retrieve (n) pointers to (n) documents hosted in the Web space. The get part of this F1 depends on how the SE search engine is structured. As in the case of Google documents and pointers (or References) to documents are indexed by words (see note 4), the argument should be a word or a string of words. In its turn this argument could be entered as a string of words separated by commas or as a string of words separated by a "blank" character and enclosed between quotation marks.

Note 4: there is an extended confusion about the use of the keyword concept within the Search Engines arena. A keyword is a symbol for a meaningful concept and as such it could be expressed in a given language by a word or by a precise string of words. A given keyword belongs to specific disciplines and for that reason issued alone without explicit reference to the discipline to which it belongs is to fall in ambiguity. As most conventional Search Engines of today are not structured by discipline they do not index by keyword but by words. We say that they are semantically "flat" in the sense that when you query by "meat" they bring you pointers to all documents that use that word within their content no matter the discipline, and for that reason you are going to retrieve answers concerning poultry meat, cow meat, horse meat, fish meat,....., steel meat, etc. However if you as a user or as an agent query precisely by the right string in "between quotations" mode you probably are going to obtain a meaningful and satisfactory answer, but it was your merit as a "connoisseur" not of the search engine.

What's then the outcome of a query?. Normally you are going to obtain from thousands to millions of references, of course depending of k (remember that k in F1 could be of the form [w1\*w2\* ....] a string of words separated by a blank space (\*)). Another consideration is that the total amount of references (#R(SE(k)) that reads: "total retrieved references for k in search Engine SE", could range from

0, between 0 and n, grater than n

#R(SE(k)) is a very important parameter because it tells you (or to your agent) the relative "abundance" of documents that mention k precisely, measuring in some extent its "popularity". F1's mission could eventually be to take only the n Top References that will have the following "conceptual" form:

Reference: [URL, r, text sample, subject, auxiliary parameters]

Where:

URL: Uniform Resource Locator, of the form [Resource type, name, domain type/path], and within it we distinguish two important variables: domain type and "extension" of the file where the content is finally stored.

r: stands for "ranking" of the document pointed, a measure of its "popularity", and we may distinguish here two kinds of popularity: the popularity within the Web owners Community and the popularity of a given document measured as the amount of users that decide (or have decided) to see it. The first one could be measured by Search Engines. The second one falls within the realm of each document privacy.

Text sample: robots that feed Search Engines may extract “ring” samples where words used to index appear circularly. Within these samples Search Engines highlight them to facilitate the references’ comprehension. These samples under the form of one or two selected paragraphs facilitate the user comprehension of the document referred.

Subject: the discipline (or sub discipline) to which the document pointed belongs. It’s not considered actually in conventional Search Engines but it should be a must in a near future.

Auxiliary parameters: information that helps the comprehension of the pointed documents such as the existence of copies within the Search Engines databases. In a near future the main keywords of pointed document should be shown (or optionally enabled their retrieval) within this section.

As you may imagine  $F1(SE, \dots)$  that means fetch in the Search Engine SE involves a complex configuration procedure, for example to know the SE structure, how are references built and presented and what kind of analysis could be performed by a human or by an agent when inspecting their content. Let’ see now the main differences when references are inspected by an agent.

References are edited by Search Engines. It means that “conceptual” reference are edited to be inspected by humans, for instance within HTML pages. The real content is camouflaged within a vast jungle of editing commands and codes.

Some Search Engines provide Web developers special API interfaces, as a filter/mask to guide agents where to find the conceptual content they are looking for. In all cases agents must locate them precisely. This could be performed by a filtering operation **Filter (SE (file))**. This operation is similar to “save as” option in txt or enriched txt format, that eliminates all the “dressing” and keep the essential.

Finally  $F1((SE, n); k)$  provides a result the should be saved for ulterior process-inspection:

$$F1 ((SE, n); k) = X$$

Being X a string of n or less references such as:

$$X = [p1|p2|p3|.....|pn],$$

With | meaning an adequate separator of references p’s of the form:.

$$p = [URL, r, \text{text sample}, \text{subject}, \text{auxiliary parameters}]$$

## **Killing Process**

We have gotten (n) or less references out of existent  $\#R(k)$ . Perhaps these n are not the best ones but at least the most important for the criteria used by Search Engine SE. However it’s perfectly possible that some p’s are useless in despite of that criteria. One way humans and agents have to evaluate this usability is by introspecting within p’s content. If we intend to optimize the search process perhaps it would be convenient to get an amount substantially higher than n, and then proceed to “kill” those considered useless for the given k. Once performed a cleansing operation of this nature the algorithm (in fact the F1 operator) may select and keep the n Top. With this procedure in three steps:

Getting  $n' \gg n$ ; killing useless; selecting the n Top,

Where killing accounts by two ways, namely: by eliminating an allegedly bad reference or indirectly by sorting the rest with a different criterion. For example if they are ranked by (r) we may proceed to sort the remaining references by another criterion such as by domain and/or by extension. By “de facto” those references that remain below the n Top cut off line once the new criteria are applied are considered as killed. This process could be formalized as

$$F1(SE, n') = X'$$

$$Kn (X', \text{ killing parameters}) = X$$

That reads: “Keep only the n Top of X’ by setting killing parameters.

At last after an appropriate killing process we get in X the n Top best references that answer query k. We may say that we have “mapped” k importance by using Search Engine SE!. Of course we did it but under a set of suppositions and restrictions.

### ***Fetch basic k's subset***

[k[0]] that belongs to a given Logical Tree

First we have to operate with a naked logical tree, only its structure and names of its branches from root to leaves. Such a tree (ontological tree) could be topologically represented by a table of two columns, codes and names. It could also be imagined as a list of pairs (code, name)

$$T = [(code, name)]$$

Code tree path = [h1.h2.....hl] for a tree of (l) levels,

We are in the beginning: a naked tree. And we need a representative sample of documents but we ignore all possible keywords and ignore the Jargon as well. Let's see how we start.

#### **Naked Tree Table example**

```

1.0.0.0*root
1.1.0.0*branch1
1.2.0.0*branch2
1.3.0.0*branch3
1.1.1.0*branch1/sub-branch1
1.1.2.0*branch1/sub-branch2
1.1.3.0*branch1/sub-branch3
1.1.4.0*branch1/sub-branch4
1.2.1.0*branch2/sub-branch1
1.2.2.0*branch2/sub-branch2
1.3.1.0*branch3/sub-branch1
1.3.2.0*branch3/sub-branch2
1.3.3.0*branch3/sub-branch3
1.1.1.1*branch1/sub-branch1/sub-sub-branch1
1.1.1.2*branch1/sub-branch1/sub-sub-branch2
1.1.1.3*branch1/sub-branch1/sub-sub-branch3
1.1.2.1*branch1/sub-branch2/sub-sub-branch1
1.1.3.1*branch1/sub-branch3/sub-sub-branch1
1.1.3.2*branch1/sub-branch3/sub-sub-branch2
1.1.4.1*branch1/sub-branch4/sub-sub-branch1
1.1.4.2*branch1/sub-branch4/sub-sub-branch2
1.1.4.3*branch1/sub-branch4/sub-sub-branch3
1.1.4.4*branch1/sub-branch4/sub-sub-branch4
1.2.1.1*branch1/sub-branch1/sub-sub-branch1
1.2.1.2*branch2/sub-branch1/sub-sub-branch2
1.2.2.1*branch2/sub-branch2/sub-sub-branch1
1.3.1.1*branch3/sub-branch1/sub-sub-branch1
1.3.1.2*branch3/sub-branch1/sub-sub-branch2
1.3.1.3*branch3/sub-branch1/sub-sub-branch3
1.3.2.1*branch3/sub-branch2/sub-sub-branch1
1.3.2.2*branch3/sub-branch2/sub-sub-branch2
1.3.2.3*branch3/sub-branch2/sub-sub-branch3
1.3.3.1*branch3/sub-branch3/sub-sub-branch1
1.3.3.2*branch3/sub-branch3/sub-sub-branch2

```

Where (\*) is a separator. The root corresponds to a discipline and branches and their corresponding nodes and leaves to its subjects. Subject names are sequences of words and primary “by default” keywords. Human Knowledge disciplines have from 500 to 1,000 subjects. In order to retrieve a meaningful sample we need a meaningful set of keywords representing the whole discipline spectrum. We have another basic need to “learn” as much as possible from disciplines: to know its “Jargon”, the technical language, the set of necessary “Common Words” to document whatever we imagine about the discipline.

A significant sample of a given discipline from the point of view of knowing as much as possible about its Jargon could be obtained via an unstructured searching with a set of basic keywords obtained extracting them from names of the naked logical tree. To be sure that we retrieve only documents belonging to the discipline at hand we query by pairs [discipline name, k], being k a keyword belonging to the basic set [k]0: keywords extracted from subjects names.

## ***E: Extract applied to T Operator***

E (T), operates over naked trees T, particularly over the second column of names to extract basic keywords. First of all we have to define the Basic Common Words subset, a few hundred of words, without any specific meaning concerning the discipline at hand, that normally act as semantic filling and make documents readable and meaningful.

$$E(T) = Y$$

Where Y is a list of possible basic single, double and triple word keywords subset built from T names, ignoring isolated words belonging to the basic Common Words subset.. That is if a subject is [w1 w2 w3 w4 w5] we consider as possible keywords to extract from it. Take into account that even though w's are common words, sequences like (ww), (www), (www..) could be potential keywords.

Once determined the basic keywords set they are sorted alphabetically within Y. Y could have the form of a double column table where keywords in the first column are listed with their respective frequency of appearance within T in the second column. Once finished the operation E over T either a human may optionally inspect Y to kill ambiguous and nonsensical potential keywords or the agent could be instructed to "cut" by a pre established frequency criterion and/or to eliminate syntactically "bad formed" sequences

## ***Fetch a significant sample of Documents***

### ***First Step: Fetch References***

F1(SE, n)(k) = X , being X a list of References, and then

$$F1(SE, n)(\text{for all } k \text{ pertaining to } Y) = Z = [Z1 U Z2 U Z3 U \dots Z(m-1) U Zm],$$

Where (m) stands for the Basic Keywords Subset cardinal and U is the conventional sets union logical operator. Z is then a sequence of Reference lists for each k pertaining to Y.

### ***Second Step: Cleansing***

Now References must be cleansed stripping them off everything but their URL's

$$C: \text{Cleansing: } C(Z) = S = [URL1 URL2 URL3 \dots URLq]$$

Where S is in fact a "Bookmark", a list of URL's of documents that represent the discipline we are analyzing. The amount of documents is (q). Now we have at our disposition either a file (for instance an exported Bookmark) with q links. We may define now another operator Get (G) to perform the capture of all those q documents grouped in files one appended to the other of a size easy to handle.

### ***Third Step: Capture Meaningful Content***

$$G(\text{size easy to handle; } Z) = \text{Folder of (rounded volume of } q/\text{size easy to handle) files}$$

The agent that performs G must process a document retrieved from the Web at a time and strip them off of their dressing as a function of their extensions. For instance the agent will face documents edited in PDF, PS, DOC, HTML, XML, PHP, ....., formats, and must proceed to strip them off their respective "dressing" keeping only their meaningful content. For that it will need the filter operator ***filter (settled up by extension type)***.

Example:

q: 500,000 documents belonging to a single discipline  
Estimated size of each document: 50 KB  
Estimated size of neat content: 5KB  
Manageable size: 1,000 KB ~200 documents content

Partitions within folder: 2,500 files

Inner structure of each file: [...#w\*w\*w\*\*\*w.....i\*\*\*w#.....] where:

(#): beginning-end of paragraph

(\*): separators, blank spaces, special symbols

(w): words

(i): images, links

Note 5: any combination of words could be potential keywords

Note 6: words should be matched against adequate dictionaries.

## ***The hidden literary pattern of a given discipline***

Any discipline has a core of “Authorities”, documents, and/or Websites that “speak” and broadcasts formally its know-how to the world. Their content is written combining literarily three semantic objects, namely:

- “Common Words” (cw’s);
- Special and meaningful concepts, keywords (k’s), normally expressed as strings of common words, and;
- Images, links, external resources, quotations, references.

Ignoring the third type of objects we have then ideas expressed by strings of words (cw’s) and keywords. The sets of Common Words and keywords have their size bound, limited, defining the discipline Jargon and Thesaurus respectively.

Being #(cw’s) and #(k’s) the cardinals of Common Words’ set and keywords’ set respectively we may state that both depend of the size of the discipline Content Sample  $q$ . We argue as a matter of common sense that both cardinals grow as a function of  $q$  following an accumulative exponential logistic curve. If we suppose in the example above that 2,500 files of content means a practical infinite it would be interesting to know the behavior of both cardinals as a function of sample size.

Let’s imagine that we have found an algorithm to separate (cw’s) and potential (k’s) in two baskets for each sample size. Being that the case we may then compute the “S curves” curves precisely. If the algorithm discriminates well between words and keywords their frequency distribution could be then computed for each sample size as well. By human experience and fostered by common sense we dare to establish as a strong conjecture that cw’s will be by far more abundant than k’s. It means that “authoritative writers” used to write documents naming keywords a few times along their discourse, in the beginning, perhaps in the middle and sometimes at the end even though in different ways and with differences in their deepness. So we have only a few keywords, the ones we need, dispersed and very diluted among hundred and thousands of common words. If that proved to be true it will be easy to recognize them analyzing within frequency distributions the expected “cuts” between cw’s and k’s.

If the starting sample was one of 200 documents we may instruct the agent to use the algorithm at exponential steps in the following sequence:

1 => 2 => 4 => 8 => 16 => 32 => 64 => 128 =>  
256 => 1024 => 2048 => 4096 => 8192  
till fulfillment of the full Sample Volume

## ***Parsing P Operator***

Experimental procedure to extract cw’s and k’s from a Discipline Sample

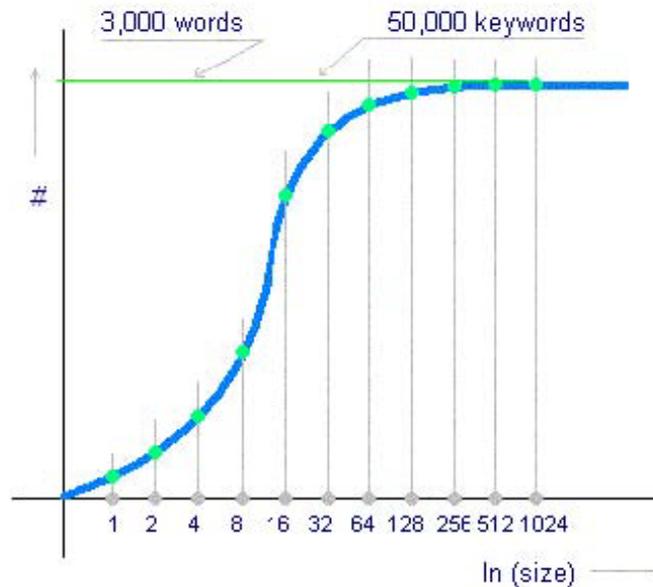
In this context we understand by parsing discrimination between common words and potential keywords. We create an Operator  $P$  that works threading words sequentially in only one sense, from left to right in packets from 1 to  $h$ . Along its threading task it match words with keywords in the basic keywords set  $[k]_0$  in order to respect them as keywords keeping from splitting them in words. Each packet is stacked and accounted for in a double column table of  $n$ -ads (from 1 to  $h$  words) with their respective abundance within the sample. It would have the form

***$P$  (i, sampling at random over Sample) giving as a result a list  $Li(k', freq)$***

That is a list of potential keywords (k's) at step i sample size (i). This list could be seen in two forms, namely, either sorted alphabetically or by frequency. We are going then to have the following sequence:

$$\# L1(k', \text{freq}) \Rightarrow \# L2(k', \text{freq}) \Rightarrow \# L4(k', \text{freq}) \dots \Rightarrow \# L8192(k', \text{freq})$$

Where the symbol (#) stands for cardinal of either k's or cw's. This process has to be considered an experiment performed by agents under the technical and scientific supervision of a human. It will be in essence an agent's settling and tune-up process. Cardinals to be controlled are two, one for k's and one for common words, taking into consideration that concerning frequencies both packet types will tend to apart from each other more and more statistically. The alphabetical list mode is used to evaluate "fuzzy zones" where elimination/transfer of eventual keywords that appear wrongly in the cw's zone or on the contrary, some single word that appears wrongly in the k's zone. Another decision to be optionally confirmed by human inspection is the "cutting" line that separates both baskets. At the end of the process it's supposed that we are going to unveil the Common Words' set (dictionary) and a potential keywords' set, even though unstructured, flat!.



In the figure above we schematize how the research is performed to obtain both baskets for a given discipline, characterized by a 3,000 Common Words' Jargon, 50,000 keywords, still unstructured, and with its content defined by a BVL, Basic Virtual Library (Knowledge Base) of nearly 500,000 documents.

## Thesaurus

A thesaurus of a Major Discipline of the Human Knowledge could be defined as the triad [d, (k, s)] where d are documents of a meaningful Virtual Library of the discipline and pairs (k, s) all possible keywords sets related to the subjects of the Major Discipline Curricula. Sometimes could be referred only as pairs [(k, s)] and sometimes confused with the whole set of keywords, configuring what we used to call a "flat" Thesaurus. Remember how we started:

$$\begin{aligned} \text{Given the Curricula} &\Rightarrow \text{a Sample of documents (d)} \Rightarrow \text{basic keywords set [k]0} \\ &\Rightarrow \text{Recover the basic writing intelligence: [wc] and suspected/potential [k]} \end{aligned}$$

We performed up to here three huge and complex semantic operations. Now we have a more ample panorama than in the beginning. We may now better the Sample because we have at hand suspected/potential [k] of the weak [k]0. For this reason it's always possible to improve the Sample by using the fetching operator again via the flat Thesaurus.

However it seems us that it is nonsensical to go farther than getting more than 500,000 authoritative documents per major Discipline. We also have at hand more elaborated procedures to improve the Sample such as keeping a Sample subset, for instance selecting the best 50,000 documents subset that maximizes "core keywords" matchmaking as we will see later once we get a structured Thesaurus.

Now we know how to discriminate common words from keywords in any document simply by killing them. The remaining essential content is supposed to be a string of keywords, something like "finger prints" ® of the document. We have, notwithstanding to face a new problem: we ignore what the subject of the document is! Why we did not use the Curricula subjects to build the Sample?: because when subjects are literary too complex, too large, references' abundance could be scarce. For this reason we use the basic keywords' set [k]0 instead. What to do then?.

A "brute force" procedure would to enforce document clusters that share keywords. It's highly probable that those clusters are closely related to subjects but it will be no trivial matter to make right assignments via agents.

A better procedure would be to build a sample fetching settled up to "catch" authorities by subjects "as they are" no matter the amount of documents retrieved. Let's have a dozen of documents for each subject. Now we may determine the common set of keywords for each subject and then match them with keywords' clusters. Another global procedure would be to determine the common set of keywords per subject "as it is". Once computed all subjects a completeness test could be performed matching the union of keywords' sets versus the flat Thesaurus.

### ***Discriminating Operator DO***

It works over the essential content of documents

DO (([cw]), [(k)];(X)), operates over a content stored in X based on a set of Common Words [(cw)], and its corresponding flat Thesaurus [(k)]. It delivers a set of keywords, the one that defines the document finger prints.

DO (([cw]), [(k)], Sample (s)), operates over a sample of documents belonging to subject (s). It delivers a union of finger prints sets. The intersection of finger prints sets defines the "nuclear keywords" of subject (s), those that are nuclear to it, related to itself, namely its fingerprint. Applying the same operation to all subjects, level by level of the hierarchy, we may compute the "core keywords", those keywords that are specific to each subject. We think that it is highly desirable to differentiate for each subject their core keywords from "intra subjects keywords". DO is an extended killing operator

### ***Fetch similar Operator***

***F2 ({SE's}, n, similarity ratio, calculation time (t)); (d),***

It fetches documents similar to a given one (d), getting the n Top from a pool of Search Engine SE's. F2 make use of a Markovian Algorithm. Ideally similar are those documents who share the same finger prints. If as a result of a first GET operation it brings less references than (n), the agent activates a Markovian process eliminating at random one keyword of the fingerprint at a time until references are equal or greater than n. Process stops when either (n) is surpassed or "similarity ratio" would turn to be lower than a fixed minimum (for example 0.70 ⇔ 70%) whichever occurs first. Once process stops the agent may continue maximizing references by restarting the random process. The only remaining limitation is the calculation time (t) that could also be settled in the setting part of the Operator, for instanced fixing it in 10 seconds. This process could be improved with weighted fingerprints.

## **Fetch by keyword family Operator**

**$F_3(SE, n, t)$ ,**

Describes a new type of fetching, references indexed by a given keyword  $k$  and its closest keywords within the subject to which all of them belong. Previously to use it this closure relationship should be determined. Dealing with documents versus keywords existence (Boolean) matrix for a given subject a logical algorithm should compute for each column ( $k$ ) the amount of successful intersections with all other columns. With this algorithm we will have for each  $k$  its “proximity” with the rest.  $F_3$  must then be settled and tuned up to GO to build these proximity vectors matrixes and then GET references to add as a function of proximity.

## **Example of a Primitive “Flat” Thesaurus Build Up**

$P\{i, \text{ sampling at random; } 8 \text{ steps binary doubling sample size } \{G \{1,000; C\{K500\{F_1((\text{Google}, 1,000); E(\text{naked } T)), \text{ for } T\text{'s pertaining to HK}\}, \text{ killing parameters by subject for HK}\}\}\} \Rightarrow \text{Thesaurus}$

And in a more abbreviated algebraic style it would be

**$P\{r; 8\{G\{1,000; C\{K500\{F_1((\text{Google}, 1,000); E(T)), HK\}K)\}\}\}$**

The source data is a HK skeleton, that is the Human Knowledge structure expressed as an Ontological Tree of nearly 200 disciplines. From this skeleton is derived as a set of naked Tree Tables (T). This complex multi agent mission should be defined for a given language. Agents’ setting and tune-up should be provided via a set of parameters for each discipline.

In this way agents may automatically retrieve all keywords, discipline by discipline. In figures we are talking of a flat Thesaurus of nearly 10,000,000 keywords, at this step “flat” only structured by discipline. In order to have it structured by subjects in as many levels as inherently exists we need to make use of DO,  $F_2$ , and  $F_3$  Operators in a process not described here but easy to imagine: From the Knowledge Base of 500,000 documents of our example we may identify the “fingerprints” of each document (their keywords). We may define another Knowledge Base restricted to subjects of the Ontological Trees to find the “core keywords” of each subject. With these core keywords we may find similar for each subject improving the subject content homogeneity and keyword specificity of the Knowledge Base. Now we have got a real Web Thesaurus.

## **Bibliography and References**

This paper intent to describe agents and multi agents tasks within the digitalized aspects of Knowledge Management. As it presupposes the validity of a set of Conjectures that backs up a new Knowledge Management paradigm we do not find bibliographic authorities that cover its full spectrum but authorities dealing with specific and everlasting disciplines like Information Retrieval, Ontology, Logic, Programming, and Web Semantics as well. In order to have a common language we refer to Foldoc Glossary. You may find a [Foldoc wizard](#) in our [Demo site](#).

### **Ontology**

1. A systematic account of Existence. 2. (From philosophy) An explicit formal specification of how to represent the objects, concepts and other entities that are assumed to exist in some area of interest and the relationships that hold among them. For {AI} systems, what exists is that which can be represented. When the {knowledge} about a {domain} is represented in a {declarative language}, the set of objects that can be represented is called the {universe of discourse}. We can describe the ontology of a program by defining a set of representational terms. Definitions associate the names of entities in the {universe of discourse} (e.g. classes, relations, functions or other objects) with human-readable text describing what the names mean, and formal {axioms} that constrain the interpretation and well-formed use of these terms. Formally, ontology is the statement of a {logical theory}.

A set of {agents} that share the same ontology will be able to communicate about a domain of discourse without necessarily operating on a globally shared theory. We say that an agent commits to ontology if its observable actions are consistent with the definitions in the ontology. The idea of ontological commitment is based on the {Knowledge-Level} perspective. 3. The hierarchical structuring of knowledge about things by subcategorizing them according to their essential (or at least relevant and/or cognitive) qualities. See {subject index}. This is an extension of the previous senses of ontology (above) which has become common in discussions about the difficulty of maintaining {subject indices}. (1997-04-09)

#### *Subject index*

It is an information resource that contains references to other resources, categorized by subject, usually in a {hierarchy}. Subject indices are not to be confused with {search engines}, which are based not on subject, but instead on {relevance}, although (1) this difference is often (possibly rightly) hidden from the unsophisticated user, and (2) future integration of {knowledge representation} into relevance ranking {algorithms} will make this a hazy distinction. (1997-04-09)

#### **Knowledge representation**

The subfield of {artificial intelligence} concerned with designing and using systems for storing knowledge - {fact}s and {rule}s about some subject. A body of formally represented knowledge is based on a {conceptualization} - an {abstract} view of the world that we wish to represent. In order to manipulate this knowledge we must specify how the abstract conceptualization is represented as a {concrete} data structure. An {ontology} is an explicit specification of a conceptualization. (1994-10-19)

1. KM, Logic and Related Formalisms, University of Kaiserslautern, by (c) 2001 Prof. Dr. Michael M. Richter, Universität Kaiserslautern, Knowledge Management, [Chapter I](#) and Chapter II. It deals with some crucial themes for our aim, namely: •Representational Adequacy: Is it possible to represent everything of interest?; •Inferential Adequacy: Can new information/knowledge be inferred?; •Inferential Efficiency: How easy (computationally) is it to infer new knowledge?; •Acquisitional Efficiency: How hard is it to formalize information/knowledge?. 7

2. [ICONS](#) , [Intelligent Content Management System](#) , Feb 2003, Rodan Systems (PL), The Polish Academy of Sciences (PL), Centro di Ingegneria Economica e Sociale (IT), InfoVide (PL), SchlumbergerSema (BE), University Paris 9 Dauphine (FR), University of Ulster (UK); It deals with Access Algorithms and Data Structures Underlying a Distributed Knowledge Base.

3. Web Epistemology, <http://webepistemology.org/main> , is a platform for studying the impact of the web on the mechanisms of collective knowledge production, organization and distribution. A major part of this project is devoted to analysing the stakes and goals of scientific research management and policies with regard to the web.

4. [Distributed Knowledge Modeling through the World Wide Web](#) , by Mildred L. G. Shaw and Brian R. Gaines, from Knowledge Science Institute, University of Calgary, Alberta, Canada T2N 1N4, {mildred, [gaines](mailto:gaines@cpsc.ucalgary.ca)}@cpsc.ucalgary.ca ,

5. Castells, M. (2001), "The Internet Galaxy – Reflections on the Internet, Business, and Society", London; Oxford University Press.

6. Cyc Faq, <http://www.robotwisdom.com/ai/cycfaq.html> , by David Whitten as Editor. Cyc is an attempt to do symbolic AI on a massive scale. It is not based on numerical methods such as statistical probabilities, nor is it based on neural networks or fuzzy logic. All of the knowledge in Cyc is represented declaratively in the form of logical assertions. Cyc presently contains approximately 400,000 significant assertions, which include simple statements of fact, rules about what conclusions to draw if certain statements of fact are satisfied (true), and rules about how to reason with certain types of facts and rules. New conclusions are derived by the inference engine using deductive reasoning.

7. Newsgroups: comp.ai.philosophy and comp.ai.nat-lang.

8. [A Directory of Search Agents](#) , a Galaxy.com set of robots, spiders and crawlers references, the actual state of the art in Conventional Web Searching.

9. [Web Hunting: Design of a Simple Intelligent Web Search Agent](#) , by [G. Michael Youngblood](#) .

10. [KWIC](#) , Keyword in Context, an old but always present Concordance concept and tool, as a parsing general utility.

11. Knowledge Management – [Digital Cloning \(KES\)](#) , Digital Informational Experts Created from Prior Personal Knowledge, where KES stands for Knowledge Expert Systems. It's a reflection about knowledge, personal and collective, and what information and knowledge management are.

12. [Enhanced Knowledge Management with eXtensible Rule Markup Language](#) , Jae Kyu Lee and Mye M. Sohn, Graduate School of Management, Korea Advanced Institute of Science and Technology, 207-43 Cheongryang, Seoul 130-012, Korea. E-mail: [jklee@kgs.m.kaist.ac.kr](mailto:jklee@kgs.m.kaist.ac.kr); [miae@kida.re.kr](mailto:miae@kida.re.kr)
13. [Actualizing Context for Personal Knowledge Management](#) , by Kenneth A. Berman and Fred S. Annexstein, Department of ECECS, University of Cincinnati, Cincinnati, OH 45221. [ken.berman@uc.edu](mailto:ken.berman@uc.edu) , [fred.annexstein@uc.edu](mailto:fred.annexstein@uc.edu) .
14. [KAON](#) is an open-source ontology management infrastructure targeted for business applications. It includes a comprehensive tool suite allowing easy ontology creation and management and provides a framework for building ontology-based applications. An important focus of KAON is scalable and efficient reasoning with ontology.
15. [Classification and Clustering](#) , extracted from Kiduk Yang's Doctoral Student Thesis, [North Carolina University](#).
16. "[El Espacio Web y la Noosfera](#)", (The Web Space and the Noosphere), by [Dr. Juan Chamero](#) and published by SCIPPOOL–Red Cientifica, a bilingual Spanish-English scientific community, 2001.
17. "Towards a New Digitalized Human Knowledge Paradigm", published by SCIPPOOL –Red Cientifica, a bilingual Spanish-English scientific community. You may see it in [English](#) at and in [Spanish](#) .
18. "Towards a New Knowledge Management Paradigm", ISSN 109-2750, presented at WSEAS, the World Scientific Engineering Academy and Society Multi Conference held in Miami, USA, April 2004, WSEAS Transactions On Computers, Issue 5, Volume 3, Page 1488. It's available for members at [www.wseas.org](http://www.wseas.org) and as an abstract in [www.wseas.com](http://www.wseas.com) .
19. "[How Case Studies Methodology embeds with continuity within the millennial Teaching Learning Paradigm](#) ", published by SCIPPOOL –Red Cientifica, August 2004, a bilingual Spanish-English scientific community.
20. Presented as a New Generation Search Engines Tutorial in the Argentine Expocomm 2004 by the IEEE Buenos Aires Chapter, held in Buenos Aires, Argentina, September 2004, You may obtain a copy of it by asking for <http://expocomm.com/argentina/> or to the lecturer [juan.chamero@intag.org](mailto:juan.chamero@intag.org) .

## **Appendix - Agents' tasks**

Agents basically have missions, safeguard and affidavit protocols. Similar to human agents should be instructed, trained, settled, tuned up, and monitored as function of missions. Missions are of the type GO, DO SOMETHING, REPORT. They perform their mission openly, publicly or on the contrary as secret agents. For instance if agents are instructed to search like humans they could be instructed either to behave as regular users (as humans) or as agents presenting credentials to the search engines control. In the first case they have to be instructed to proceed like a regular user (in certain extent deceiving the search engine) and to "read" the outcomes of queries as humans do. In the second case the human that acts as the agents' master previously must establish a special agreement with the search engines administrator in order to query "legally" and openly via special API's.

An Agent is "born", authorized to "live", suspended, and eliminated by human decision. However once authorized to live it could perform the following tasks (within pre-established limits):

- It notifies, by itself
- It activates, by itself
- It deactivates, by itself
- It adjusts, by itself
- It auto generates, by itself
- It nurtures, by itself
- It unload wastes, by itself
- It perpetuates, by itself
- It reproduces, by itself
- It reports
- It establish contacts

It requires information, attention  
It executes embedded processes  
It waits  
It leaves meaningful tracks  
It recognizes, objects and patterns  
It makes inferences  
It senses

Basically

***Agents React***  
***Humans Deliberate***

By Human purpose they may interact with the World, with its “master”, and with other agents. However some thinkers are confident that agents may evolve towards: a) To have its own processes; b) To have its own Cosmo vision, its own “beliefs”, “wishes” and “purposes”.

As a colophon we have our own experience working in this project with agents and humans at the same time and in tighten interrelationship besides. In fact we started our works mapping knowledge with humans aided by agents and conventional scripts and utilities. In a second step we dare to make agents a little smarter trying to transfer them some human abilities like “discovering” keywords in documents and clusters within high order sparse matrices and astonishingly we succeed: agents could do those “tasks apparently reserved only for human talent” of course faster but better than humans in terms of quality! . .