Associative Conceptual Space-based Information Retrieval Systems

Martijn Schuemie

Dept. Information systems, room 2.007
Delft University of Technology
P.O. Box 356, 2600AJ, Delft

Jan van den Berg

Dept. of Computer Science, room H4-29
Erasmus University Rotterdam
P.O. Box 1738, 3000DR, Rotterdam

The Netherlands

Email: m.j.schuemie@twi.tudelft.nl, jvandenberg@few.eur.nl

Abstract

In this 'Information Era' with the availability of large collections of books, articles, journals, CD-ROMs, video films and so on, there exists an increasing need for intelligent information retrieval systems that enable users to find the information desired easily. Many attempts have been made to construct such retrieval systems, including the electronic ones used in libraries and including the search engines for the World Wide Web. In many cases, however, the so-called 'precision' and 'recall' of these systems leave much to be desired.
In this paper, a new AI-based retrieval system is proposed, inspired by, among other things, the WEBSOM-algorithm. However, contrary to that approach where domain knowledge is extracted from the full text of all books, we propose a system where certain specific meta-information is automatically assembled using only the index of every document. This knowledge extraction process results into a new type of concept space, the so-called Associative Conceptual Space where the 'concepts' as found in all documents are clustered using a Hebbian-type of learning algorithm. Then, each document can be characterised by comparing the concepts as occurring in it to those present in the associative conceptual space. Applying these characterisations, all documents can be clustered such that semantically similar documents lie close together on a Self-Organising Map. This map can easily be inspected by its user.

1 Introduction

The availability of huge collections of books, CD-ROMs, video movies, articles et cetera in modern libraries or their respective depositories, creates the need for intelligent search systems. There already exist many types of Information Retrieval systems (IR-systems) for that purpose. They appear to apply different levels of Artificial Intelligence (AI). It would be fine if the performances of the various systems could be compared easily. This turns out to be a difficult task, however. Among other reasons, performance is a complicated notion with many points of view. Besides comparing issues like the degree to which the collection of items is covered or their user friendliness (measured in terms as response time, and the simplicity and speed of the user interface), IR-systems are often mostly evaluated with respect to their *recall* and to their *precision*. Recall is defined as the part of the relevant information available that has actually been found, and precision is the part of the information found that is really relevant to the user [1]. Preferably, an IR-system precisely finds all items relevant to the user and nothing more. Unfortunately, the notion of *relevance* is not unique. On the contrary, it turns out to be quite personal which severely complicates a good

evaluation with respect to precision and recall. So, creating an IR-system that is able to satisfy the personal information needs of all its users, is certainly no sinecure.

1.1 Classical information retrieval systems

Every IR-system needs a type of query to do its job. The system will analyse the query, will next perform a certain type of search and will finally offer the results of this search to the user. In 'Boolean retrieval systems' [2], the user can specify a query by summing up the words which should occur - and sometimes also the words which should *not* occur - in the title or body of the documents. In some cases, logic operators like 'and' and 'or' can be used to improve the formulation of the query. The underlying system should have collected various data from the individual books like title, authors name, editor, keywords, or a summary. By comparing the words of the query to the collected data of the individual books, the search result is found. A well-known method here (also in use on the World Wide Web by the search engine Alta Vista) is that of 'inverted indices': the collected keywords of the books are placed in one huge index where, for any keyword, the occurrences are maintained. Given the query, the retrieval system may restrict its activities to an inspection of the inverted index only. A performance evaluation of Boolean retrieval systems shows a well-known trade-off between precision and recall: the more keywords are specified in the query, the higher the precision usually is. However, this high precision is usually attained at the expense of the recall. Similarly, a high recall can be obtained by using few key words at the expense of the precision quality. A deeper analysis of the often poor performance of Boolean IR-systems clarifies that such systems have little intelligence: the meta-information on the documents simply consists of a set of individual keywords where the mutual semantic relationship between these words or the underlying common notions represented by these words, is completely ignored. Therefore, a better performance may be expected for retrieval systems that take *semantic aspects* into account.

1.2 New approaches

A first improvement can be achieved by application of so-called 'word root reduction' [2]. Here, words are reduced to their grammatical root before comparing. The success of this approach is limited however and also depends on the language used in the documents.

The semantics of individual words can also be applied using a so-called 'thesaurus', a vocabulary where synonymous and semantically covering or otherwise related words are being collected. In addition to the user-given key words, the inverted index can also be inspected using the words related by the thesaurus. An obvious disadvantage of this approach is that the construction of a thesaurus in a specific domain requires a lot of specialised knowledge making it difficult to construct such a vocabulary automatically.

A different approach to improve IR-systems is the so-called *vector-space model*. Here, each document is represented by a high-dimensional vector, every component of which represents a word in the vocabulary of the text where each components value is related to the number of times the word appears in the document. Comparison of these document-vectors to a (similar) vector constructed from the query, results in a set of documents presented to the user. The largest disadvantage of this method is the fact that still no semantic aspects are taken into account.

Another development is the use of *relevance feedback*, where the user can report to the system whether the documents found are relevant to the user. Hereupon, the system can try to find documents similar to these relevant documents. This method can be used as an addition to any existing IR-system.
Many IR-systems that take semantic aspects into account, have proven to be rather time-consuming with respect to their construction because usually, the semantic knowledge (like, for example, a thesaurus) had to be created manually. They are therefore expensive. Instead, we would like to make a system that has some form of Artificial Intelligence (AI) where the necessary semantic knowledge is created automatically.

Several IR-systems that apply some form of AI already exist. One of these is the *n-gram method* [3] where documents are represented by a high-dimensional vector, each component of which represents a *n*-letter combination (usually *n=3*). The value of such a component depends on the number of times the combination of letters occurs in the text. Using a multi-layered feedforward network with error-backpropagation, the system is trained to divide the documents into certain predetermined categories. The user can find relevant documents by searching for the category that appears most relevant to him. Disadvantages of this system are the fact that, even though most word-roots are detected, still very little semantics is taken into account. Also, the categories have to be determined manually in advance. A solution to the latter problem could be to use an unsupervised learning method such as recurrent networks, but this approach has not been very successful due to the enormous increase in time-complexity [3].
Another approach is the use of Self-Organising Maps (SOMs), the correponding method of which will be described in more detail in the next paragraph. Here, the basic idea is to create a 'semantic fingerprint' of each document using the entire text-corpus. As far as we know, this method has been used on small documents only. If this method would be tried with books, we foresee a problem: books often contain hundreds of pages making it hard to analyse the entire text of each one. So, for books there exists a 'complexity problem': a book contains too many words making it difficult to construct a semantic fingerprint. Instead, we need 'more intelligent' systems that are able to construct such fingerprints using less information. An important condition for success of our approach is the assumption that in the near future, certain information about the books will be available on an electronic medium, at least in the offices of the publisher. Besides title, author, date of publication, et cetera, this electronic information should include the *index* (and, if desired, the table of contents as well) of the book which seem to contain the necessary semantic information to make a significant fingerprint. The question now becomes how to exploit this, in volume rather limited, information. Solving this problem is the goal of this paper.

1.3 Outline of this paper

In paragraph 2, we will look at two existing systems that are capable of constructing domain-knowledge from full-text document representations. Using this analysis, a new general architecture for 'index-based' systems will be derived. In the next paragraph, two algorithms will be introduced that implement an IR-system having such an architecture. In the last paragraph, we draw some conclusions and present an outlook on the future.

2 Knowledge based IR-systems

Already a number of systems exist where attempts were made to use domain-knowledge to gain a higher precision and higher recall than is commonly attained using the traditional systems described in the previous paragraph. Two such systems are the WEBSOM [4] and the Aqua-browser [5], the last one of which is an IR-system based on Connectionist Semantic Networks.

2.1 The WEBSOM-algorithm

The WEBSOM-algorithm uses full-text documents as input. After having removed all non-alphabetical characters and less frequent words (e.g. words that occur less than 50 times in the text), each remaining word is represented by a unique *n*-dimensional real vector x_i (e.g. $n=90$) with random-number components, where *i* denotes the *i*th word in the text. The relation between the words is determined using the average short context. Using the vectors x_i the average context vector of each word reads

$$X(i) = \begin{bmatrix} \mathrm{E}\{ x_{i-1} \mid x_i \} \\ \varepsilon\, x_i \\ \mathrm{E}\{ x_{i+1} \mid x_i \} \end{bmatrix},$$

where E denotes the estimate of the expectation value evaluated over the text corpus, and where ε is a small scalar number (e.g., $\varepsilon = 0.2$). The purpose of ε is to reduce the influence of the (randomly determined) vector of the keyword: a relatively small value of ε increases the focus on the context of the word.

The $X(i) \in \Re^{3n}$ constitute the input vectors to the so-called *word category map*. This map is a Self-Organising Map (SOM) [6], a neural network-based unsupervised tool for ordering high-dimensional statistical data such that - in general - alike inputs are mapped close to each other. Using the X_i (with different word vectors x_i) as a training set for the SOM, words x_i with similar context vectors will be mapped close together. The word category map has far less units then there are words present in the text, forcing the SOM to represent several words with only one unit. In other words, the SOM clusters words with similar contexts. Using the word category map, a fingerprint of every book is constructed consisting of a clusters histogram. To do so, each word occurring in the text is assigned to the cluster it belongs to conform the category map. The histograms of all documents are then used as input for a second SOM, the so-called *document map*. On this map, documents that address similar topics are, in general, mapped close together. See also figure 1.

2.2 Connectionist Semantic Network-based algorithms

IR-systems based on Connectionist Semantic Networks (CSN) [12] usually also start out removing less frequent words and non-alphabetical characters. The remaining words are placed in a network where each node represents a word and the weighed connections between the nodes represent the strength of the relations between the words. Normally, these weights are based on word-co-occurrence: words that often appear close to each other in the text are

more strongly connected. This CSN can then be made accessible to the user by means of a graphical interface such as the Aqua-Browser, where words can be selected using a mouse-click. Next, words related to this selection are made visible and selectable. After the user has identified the key concepts (s)he is interested in, the system attempts to find all documents that match these requirements. This can be achieved by using a traditional Boolean search method with the words selected by the user, augmented with the words closely related in the CSN, as the search-string.

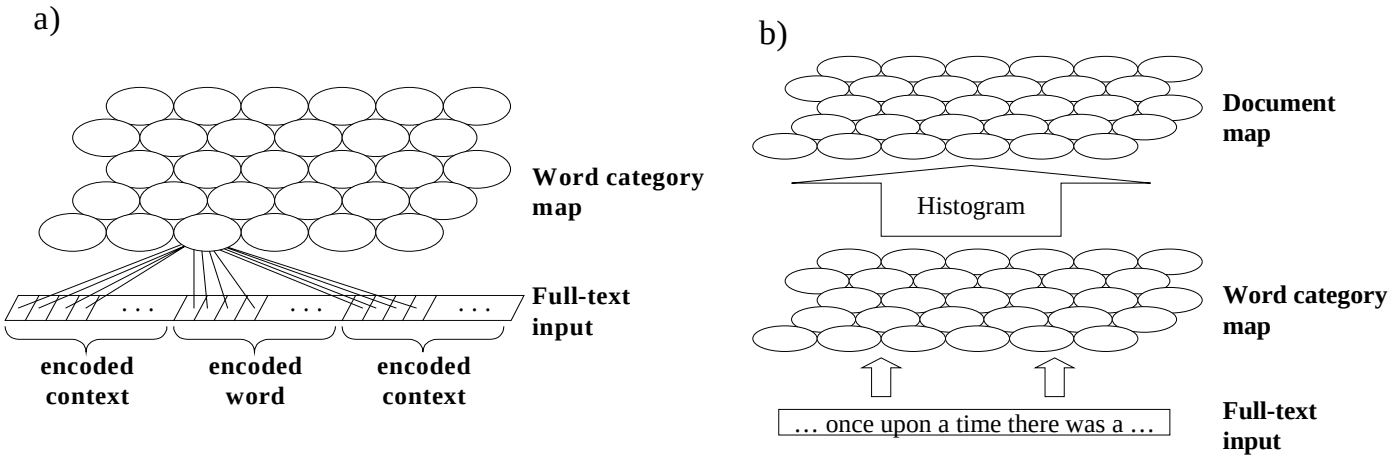

Figure 1:The basic two-level WEBSOM architecture. (a) The word category map first learns to represent relations of words based on their average contexts. This map is then used to form a word histogram of each document to be analysed. (b) The histogram, a 'fingerprint' of the document, is then used as input for the second SOM, the document map.

2.3 A new architecture

Both systems just briefly reviewed, show a new development in IR-systems: instead of matching documents on a word-by-word basis using the words occurring in the query, the documents are analysed to find underlying 'concepts' to which these words are related. We could also say that these systems attempt to compare the documents and the query given on a higher level of abstraction. This idea is visualised in figure 2. An important part of an IR-system having such an architecture, is the method used to extract the relevant domain knowledge from the available document representations. Both systems described in this paragraph, attempt to build this knowledge base by finding relations between words in the text. Little or nothing has been said however about the nature of these relations. Two types of relations between words can be distinguished [7]:

1. Semantic relations: this concerns words having to a certain extent similar meanings, such as 'bread' and 'cake'.
2. Associative relations: this concerns words that are 'associated' (but do not need to have similar meanings), such as 'key' and 'lock'.

Using the short-context method of the WEBSOM, one would expect to find mostly semantic relations because words with more or less similar meanings tend to be used in the same place in a sentence. (E.g. 'eat your BREAD now' or 'eat your CAKE now'). The word-co-occurrence method of CSN-based IR-systems however, will most probably find

associatively related words, because these words tend to be used in each other's vicinity, often in the same sentence. (E.g. 'LOCK the door with the KEY' or 'the KEY wouldn't fit in the LOCK').

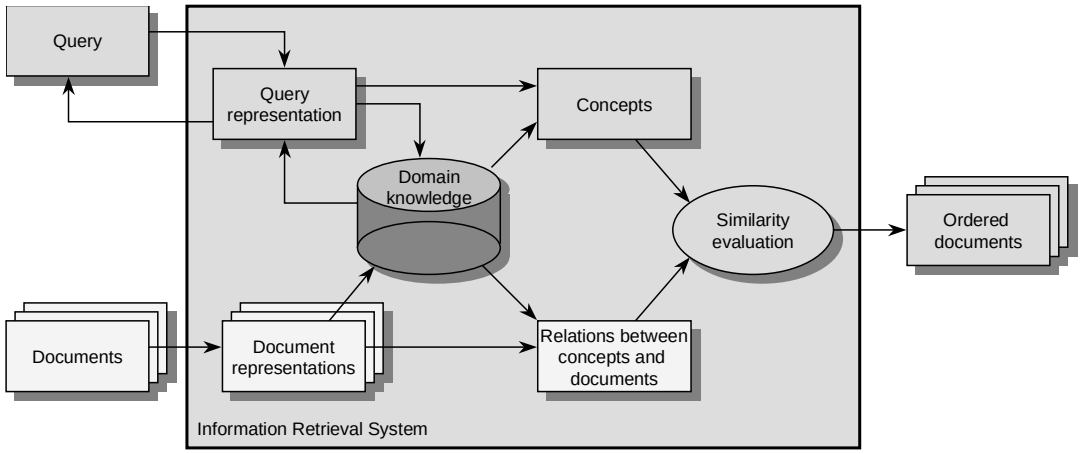

Figure 2: Architecture of an IR-system that uses domain-knowledge to match documents and query on a higher level of abstraction. This knowledge base can then also be used to aid the user in formulating the query through an interactive (and usually graphical) interface.

In order to match documents against a query, we must have a method which allows us to relate the documents to the concepts found during knowledge extraction. The WEBSOM uses a holistic approach, creating a histogram of the concepts addressed in a document. CSN-based IR-systems map the words occurring in a document to the concepts addressed in a query.

The concepts describing a document have to be extracted from the available internal representations of the documents. Until now, we assumed that full-text representations of all documents are available and that these representations are small enough to be handled within a realistic timeframe. This, however, is not always the case. In libraries for instance, we find (very) many books of several hundred pages which are not available in electronic format. Imaging is costly and the optical character recognition process needed is far from flawless. Even if these books were available in electronic format, their sheer size would make it hard, if not impossible, to process and analyse them on a word-by-word basis. To make these books accessible, it would be preferable to use only a small part of the entire book, such as the *index*. In the next paragraph, this appraoch is adopted. The corresponding method which uses a new form of knowledge-representation, is dubbed Associative Conceptual Space (ACS). The resulting complete IR-system, we named the ACS-WEBSOM.

3 ACS-WEBSOM

At first sight, an index seems to hold very little information on the meaning of the words occurring in documents. Only a few words from the original text are represented and, to make matters worse, their order has been replaced by an alphabetical one. An advantage on the other hand is the fact that the words present in the index, have been found significant enough to be mentioned[1]. Furthermore, part of the order in which these words appear in the text can be reconstructed using the page-numbers as depicted in figure 3. For each page the words can be determined that apply to that page according to the index. It is still unclear what the order of the words is on a single page but we could hereby make the assumption that *if words often occur on the same page they most likely are related*.

If we assess the usability of the knowledge-extraction methods described in the previous paragraph, we can conclude that the nature of this 'converted index' makes it impossible to use the short-context method of the original WEBSOM: too little information about the original text-corpus remains to determine the way in which the word at stake has been used in a sentence, or, stating this differently, which words occur in the short context cannot be derived from the index alone.

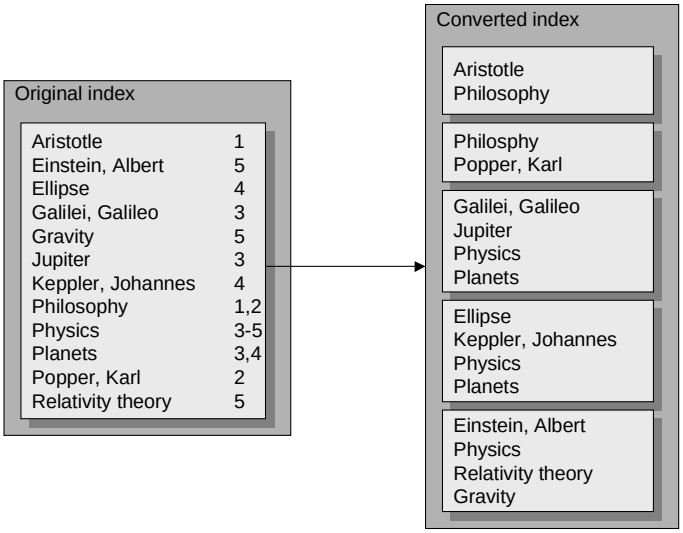

Figure 3: Conversion of an index

For the word-co-occurrence–method used by CSN-based IR-systems however, the information available in such a 'converted index' will suffice. From the index, we can deduce which words appear on the same page and therefore which words appear in each other's vicinity. It then is logical to use this method of knowledge-extraction, making CSN based IR-systems an obvious choice.

Books on the other hand, often contain a long array of concepts and it could be possible that the holistic coupling of documents to concepts as used by the WEBSOM, performs better in describing the contents of those documents. Here, we describe an approach that applies the original WEBSOM in a new way, although we admit that CSN-based IR-systems might be just as effective. For now, we will restrict ourselves to presenting this new method and leave the comparing for future research.

The problem of using the original WEBSOM-method with indices lies in the fact that WEBSOM uses vectors as input, whilst the word-co-occurrence method is based on a network architecture.

[1] Valuing the contents of an index is a matter of the subjective opinion of the author.

3.1 Associative Conceptual Space

A collection of vectors defines a space and a solution to the problem could be to create a conceptual space using word-co-occurrence. Since we established that word-co-occurrence tends to find associative relations between words, we will dub this space the Associative Conceptual Space (ACS). This space contains a collection of n vectors X, $X = \{x_1, x_2, x_3, \ldots, x_{n-1}, x_n\}$ where each vector consists of d components, depending on the number of dimensions in the space, so $x_i = \{x_{i\,1}, x_{i\,2}, x_{i\,3}, \ldots, x_{i\,d-1}, x_{i\,d}\}$. Each of these vectors is one-to-one related to a word in the collection W, $W = \{w_1, w_2, w_3, \ldots, w_{n-1}, w_n\}$. This one-to-one relation of course is a simplification, for one reason because it ignores the existence of homonyms. But it will make things a lot easier[2]. In this conceptual space, relations between words cannot be represented by connections. Hence, we need a new metric for the strength of a relation: *the distance between the reference-vectors indicates the strength of the association between the concepts related to those vectors.* We can write this as:

Association between w_i *and* $w_j = || x_i - x_j ||$ (where $\| . \|$ returns the Euclidean distance).

In order to represent the correct associations, a learning process is needed. For this, we can use the Hebbian rule of learning [9]. If two concepts are simultaneously activated[3], the strength of the connection between these concepts has to be increased. In terms of our conceptual space: *reference vectors of concepts that are activated simultaneously have to be brought nearer.* Or, to be more precise

if w_i and w_j are activated, then $|| x_i(t+1) - x_j(t+1) || < || x_i(t) - x_j(t) ||$.

To achieve this, vectors have to be adjusted. In order to make sure that each association is treated equally such that 'long-distance' adjustments do not destroy a tediously constructed 'short-distance' ordering, the adjustments should be normalised. The association-rule thus reads:

$$x_i(t+1) = x_i(t) + \eta(t) \frac{x_j(t) - x_i(t)}{|| x_j(t) - x_i(t) ||}$$

$$x_j(t+1) = x_j(t) - \eta(t) \frac{x_j(t) - x_i(t)}{|| x_j(t) - x_i(t) ||}$$

The difference between the vectors is normalised and then multiplied by the learning-rate $\eta(t)$.

[2] It is interesting to note that children when learning to speak also tend to prefer this one-to-one relationship [8].

[3] Which concepts should be activated will be discussed later on.

3.2 Active forgetting

Simply applying the given association-rule alone may easily lead to false associations, as portrayed in figure 4:

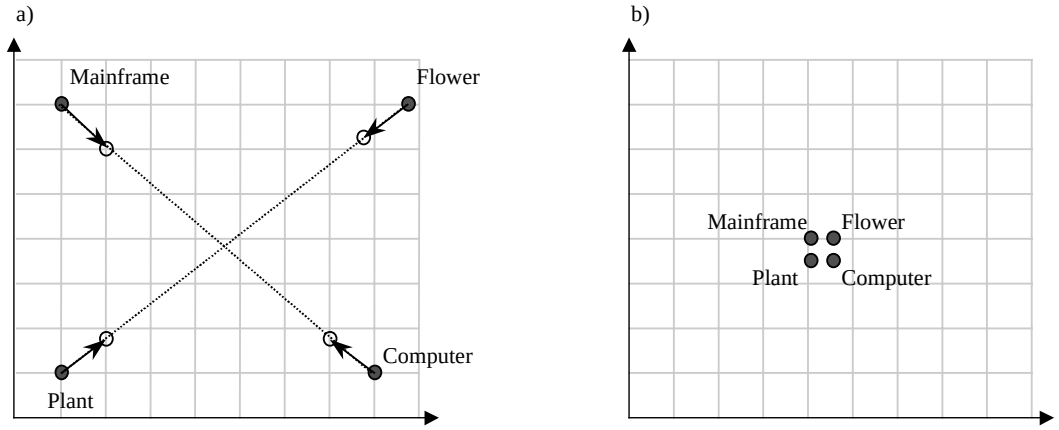

Figure 4: a) From the initial position, related concepts are brought together; b) At the same time, the distance between unrelated pairs of concepts is decreased as well, creating a false association in the conceptual space.

In this example, we see that the distance between the two related concepts 'mainframe' and 'computer' has indeed decreased, as is also the case for the pair 'plant' and 'flower'. An unwanted side effect is that these two unrelated pairs of concepts are also brought closer together: a false association is created. To counter this effect a type of *active forgetting* must be introduced that increases the distance between unrelated concepts. We could simply increase the distance between all the reference vectors using the *forget rule*:

$$x_i(t+1) = x_i(t) - \lambda\,(delta)\ \frac{x_j(t) - x_i(t)}{delta}$$

$$delta = \| x_j(t) - x_i(t) \|$$

Here again, the adjustment is normalised to counter extreme 'long-distance' effects. In addition, the adjustment is also made dependant of the distance through the function $\lambda(delta)$. If the effect of the active forgetting is mostly local, (false) associations will still be targeted while correct orderings in other parts of the conceptual space will remain intact. Therefore, $\lambda(delta)$ should be decreasing over *delta*, say $\lambda(delta) = 1/delta$. If $delta \rightarrow 0$ however, we see that $\lambda \rightarrow \infty$. We could solve the resulting erratic behaviour by making λ constant within a certain range:

$$\lambda\,(delta) = \begin{cases} 1 & \text{voor } delta < 1 \\ 1/delta & \text{voor } delta \geq 1 \end{cases} \qquad \text{(repulse function)}$$

In figure 5 we see some of the dynamics we could expect from such a combination of learning through association and active forgetting. In figure 5a, we see that a concept could take a 'detour' around unrelated concepts to a concept that it is related to, avoiding false associations. This requires enough space to make such a detour, which brings us to the dimensionality of the conceptual space. More dimensions will give the concepts more room to manoeuvre and will enable the creation of a suitable ordering. In figure 5b, we also see that if a concept is related to two other concepts which are not themselves related, an ordering will emerge where the distance between these unrelated concepts is maximised while at the same time the distance between the related concepts is minimised. Here, a higher dimensionality will enable the network to represent more unrelated concepts which are related to a single other concept without creating false associations. The dimensionality should on the other be minimised to reduce the computational load of the algorithm.

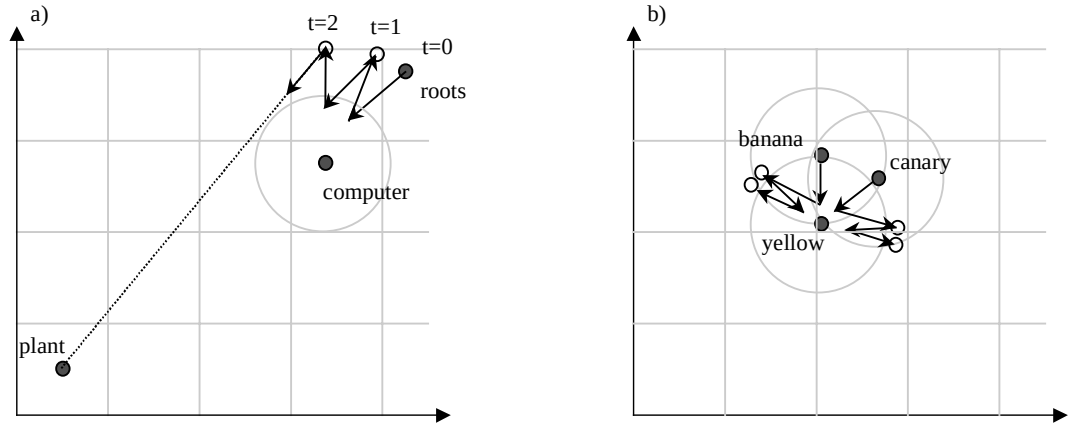

Figure 5: Dynamics in the ACS. The circles represent the repulse-behaviour of the active forgetting.

Experiments show that an initial learning-rate $\eta(t)$ in the association-rule that overshadows the forget-rule in scale will also help speed up the forming of a suitable ordering. The learning-rate can decrease during training so later adjustments will take the form of fine-tuning. On the other hand, a learning-rate which is too low will result in the destruction of the ordering by the active forgetting . So, the learning-rate should not fall below a certain minimum.

3.3 ACS from an index

To form an ACS from an index, this index should be converted as explained earlier. The words in this converted index should be concatenated into a single text-string T_i with m index-terms t_j , $T_i = \{t_{i\,1}, t_{i\,2}, t_{i\,3}, \ldots, t_{i\,m-1}, t_{i\,m}\}$ where i denotes the ith book in the collection of k books. These strings can then be concatenated into one string T_* where the books can be placed in any order: $T_* = \{T_1, T_2, T_3, \ldots, T_{k-1}, T_k\} = \{t_{1\,1}, t_{1\,2}, t_{1\,3}, \ldots, t_{k\,m-1}, t_{k\,m}\}$. We now define W as the *vocabulary* of the entire text-string, so that each element w_l of W is unique and $\forall\, w_l \in W \rightarrow w_l \in T_*$. For each word t_{*c} in the text-string T_* we can define a neighbourhood N_c with radius r, $N_c = \{t_{*c-r}, \ldots, t_{*c-2}, t_{*c-1}, t_{*c+1}, t_{*c+2}, \ldots, t_{*c+r}\}$. Each element of the neighbourhood can then be activated in combination with t_{*c}.

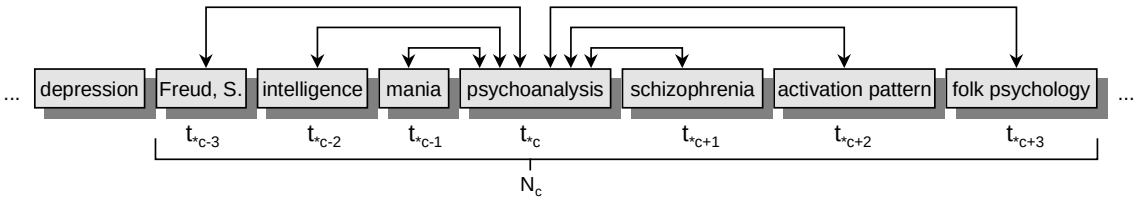

Figure 6: Example of a neighbourhood with *r=3* and the activation combinations made.

We can now train the conceptual space by going through the entire text-string T_i several times whilst also applying the forget-rule.

The frequency with which words occur can also be taken into account. Words that occur very often have a lower informative value. The association-rule should then take this form:

$$x_i(t+1) = x_i(t) + \eta(t) \frac{x_j(t) - x_i(t)}{|| x_j(t) - x_i(t) ||} \cdot \alpha(i,j) \qquad \alpha(i,j) = \frac{2.freq_*}{freq_i + freq_j} \qquad freq_i = \text{frequency of word } i$$

$$x_j(t+1) = x_j(t) - \eta(t) \frac{x_j(t) - x_i(t)}{|| x_j(t) - x_i(t) ||} \cdot \alpha(i,j) \qquad freq_* = \frac{|T_*|}{|W|} \qquad |\,.\,| \text{ returns the number of elements in a collection}$$

Hypothesis: The ACS-algorithm will result in a conceptual space C_s where points in this space represent words and where words that are closely associated in the text, are in each other's proximity in C_s.

3.4 Overview

Figure 7 shows an overview of the ACS-WEBSOM-algorithm. First, the indices of the books are converted. Words that do not occur very often, are then removed from the resulting text-string. Using the ACS-algorithm, a conceptual space C_s is created. Our experiments showed that a five-dimensional space tended to give the most satisfying ordering, although this is most likely dependent on the complexity of the knowledge to be represented.

After application of the ACS-algorithm, C_s should contain a multidimensional knowledge structure. To use this knowledge, we can apply the WEBSOM-algorithm; first, a word category map is created based on this space. By simply using the reference vectors of C_s as training vectors for a SOM, we can create a two-dimensional map of the knowledge structure as shown in figure 8.

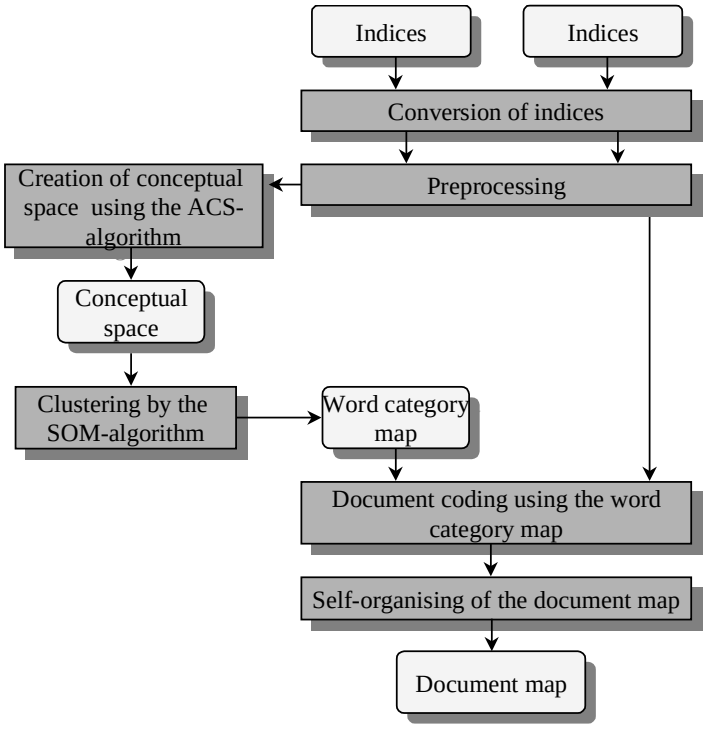

Figure 7: Overview of the ACS-WEBSOM-algorithm

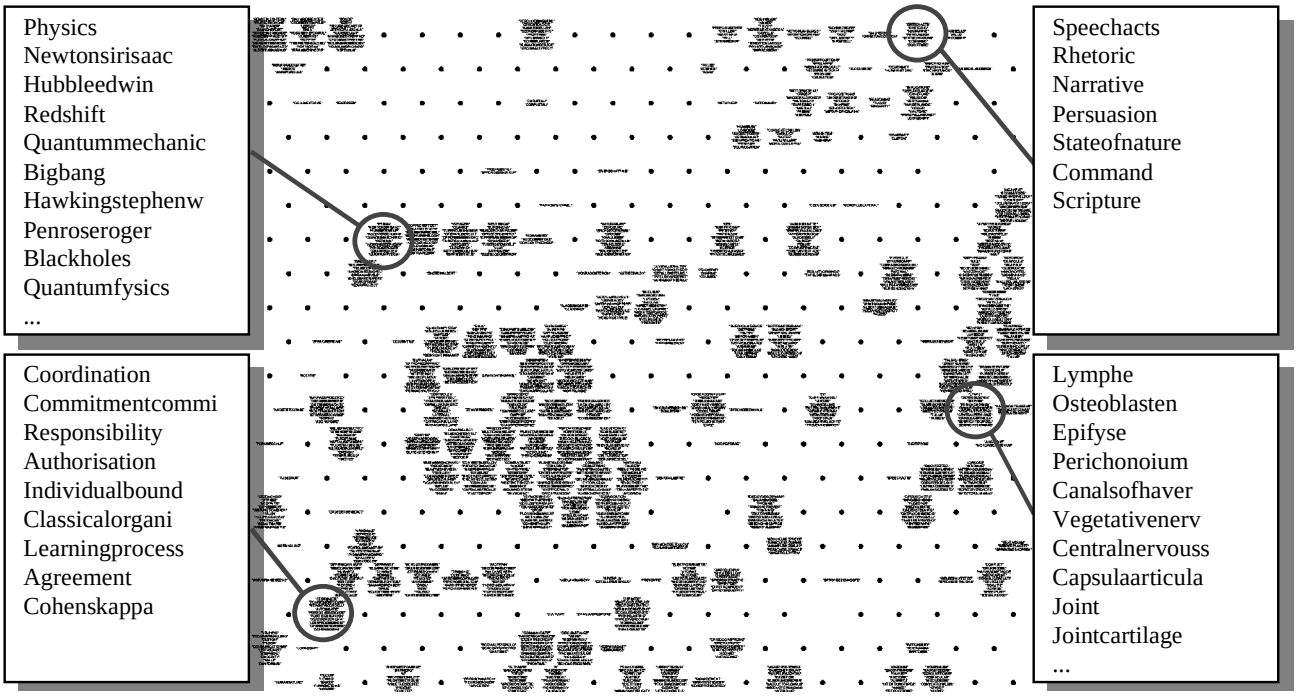

Figure 8: Examples of some clear categories in a word category map. The map was created using 40 indices and using 250 training-steps for the ACS-algorithm, after which the SOM-algorithm was applied to create the two-dimensional map of the 5 dimensional conceptual space.

Now, similar to the original WEBSOM, the word category map is used to create a document map as is depicted in figure 9.

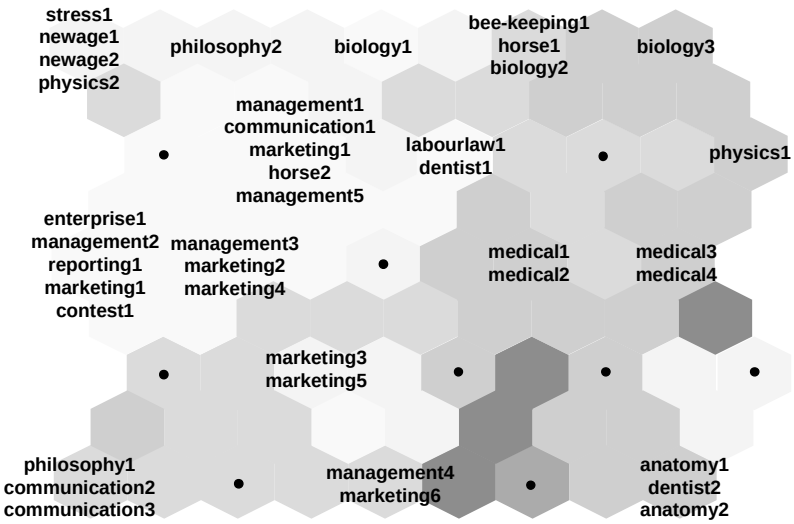

Figure 9: Document map based on the word category map of figure 8. The documents have been manually labelled according to a crude categorisation. Note that most books from the same category are found close to each other on the map.

4 Conclusions and outlook

The ACS-algorithm seems to be capable of extracting domain knowledge from documents using an Associative Conceptual Space. Experiments having results such as depicted in figure 9, show that the algorithm is able to create an ordering which is recognisable by and acceptable to humans. Further evidence shows that the algorithm will create very similar orderings independent of the initial values for the reference vectors. Here, the ACS-WEBSOM-algorithm differs from the WEBSOM-algorithm in an important aspect: The context-vectors of the WEBSOM contain a lot of random components, which will have its consequences for the resulting clustering in the word-map. Whether this means that the ACS-WEBSOM has a better performance than the original WEBSOM, is still unclear and is also hard to determine since the two algorithms differ in more aspects. To definitely decide on this, further research is needed.

The biggest problem at this moment is the time-complexity of the learning process. This is mainly due to the application of the forget rule, which takes up approximately 95% of the processing time, even if the rule is applied only once every learning-step. This way processing a document-set of 40 indices with a vocabulary (after pre-processing of 2.439 words and a text-string of 20.621 words), took 8 hours and 10 minutes on a pentium-133 system. No attempt has been made however, to optimise the algorithm because the experiments until now have been of an explorative nature only.

The ACS-WEBSOM-algorithm describes an IR-system that compares documents based on concepts constructed from word-clusters instead of based on individual words. In this way, the system is expected to achieve a higher precision and recall than traditional IR-systems, although this hypothesis has still not been proven.

Further research on this subject should first of all be focused on reducing the time-complexity of the algorithm. Also the possibility for using the ACS-WEBSOM-algorithm on full-text documents should be investigated, so that an IR-system can be created that is able to search based on both the index and the text-corpus itself of the present books, allowing the user an access to a wide variety of documents.

Acknowledgement

We thank Marc Leipoldt for reading this article and suggesting some changes to the English.

Bibliography:

[1] C.J. van Rijsbergen, "*Information Retrieval*", London: Butterworths, 1979, http://www.dcs.glasgow.ac.uk/Keith/Preface.htm

[2] R. Ferber, "*Information Retrieval*", GMD-IPSI, http://www.darmstadt.gmd.de/~ferber/ir-bb/frame.html

[3] J.C. Scholtes, *"Neural Networks in Natural Language Processing and Information Retrieval"*, PhD thesis, University of Amsterdam

[4] S. Kaski, T. Honkela, K. Lagus and T. Kohonen, "Creating an Order in Digital Libraries with Self- Organizing Maps", in "*Proc. WCNN'96 World Congress in Neural Networks"*, pp. 814-817, Lawrence Erlbaum and INNS Press, Mahwah, NJ, 1996

[5] W.A. Veling, "The Aqua Browser: Visualisation of large information spaces in context", in "*AGSI journal*", November 1997, Vol. 6, Issue 3, pp. 136-142

[6] T. Kohonen, "*Self-Organizing Maps*", Springer, 1995

[7] D.C. Plaut, "Semantic and Associative Priming in a Distributed Attractor Network", in "*Proceedings of the 17th Annual Conference of the Cognitive Science Society*", pp.37-42, Hillsdale, NJ, Lawrence Erlbaum Associates

[8] M. Imai, D. Genter, "A cross-linguistic study of early word meaning: universal ontology and linguistic influence", in "*Cognition*", nr. 2, feb. 1997, pp.196

[9] D.O. Hebb, "*The Organization of Behavior: a Neuropsychological Theory*", John Wiley & Sons inc., 1966

[10] M.J. Schuemie, "*Associatieve Conceptuele Ruimte: een vorm van kennisrepresentatie ten behoeve van informatie-zoeksystemen*", Master thesis, 1998, Erasmus University of Rotterdam http://www.few.eur.nl/few/people/jvandenberg/masters.html

[11] T. Honkela, S Kaski, K. Lagus and T. Kohonen, "*Newsgroup Exploration with WEBSOM method and Browsing Interface*". TKK Offset 1996

[12] L.Shastri, "*Semantic Networks: An Evidential Formalization and its Connectionist Realization*", Pitman Publishing, 1988