ISSN 1018-5593

1995

LIBRARIES in the INFORMATION SOCIETY

Artificial neural networks for information retrieval in a libraries context

European Commission, DG XIII-E3

EUR 16264 EN

1995

LIBRARIES in the INFORMATION SOCIETY

Artificial neural networks for information retrieval in a libraries context

Author:
Dr ir Johannes C. Scholtes

European Commission, DG XIII-E3

EUR 16264 EN

Published by the
EUROPEAN COMMISSION

Directorate-General XIII
Telecommunications, Information Market and Exploitation of Research
L-2920 Luxembourg

Cataloguing data can be found at the end of this publication

Luxembourg: Office for Official Publications of the European Communities, 1995

ISBN 92-827-4690-9

Printed in Italy

Table of Contents

10 Filter Prototype

-- Marco-René Spruit

Introduction

The approach taken in FILTER consists of matching incoming full-text data, such as news and abstracts, to a neural network representing a specific user interest. Only data correlating with this interest is returned to the user. This is known as selective dissemination of information (SDI) or the filter principle.

The amount of information which is stored in libraries is growing exponentially. This exponential growth makes it virtually impossible to maintain a manually structured database for all incoming data. Also, information storage sources are moving from analogue form towards digital form. These shifts make the automatic signalling and distribution of incoming information by computer services respectively increasingly important and more generally applicable. This seems an obvious task for libraries, being genuine information archivers.

The present chapter is divided into three parts. First, a basic theoretical background is provided to explain the context of the project. Next, the prototype is described by reviewing the implementation of some generally important application properties and by an example of an imaginary session. Finally, it is explained how the prototype has been evaluated and what conclusions can be drawn, based on this evaluation.

This entire chapter is also available within the FILTER prototype itself as part of the on-line documentation.

10.1 Background

Information retrieval

Information retrieval is the matching of a query against a large number of documents. Two types of application environments can be distinguished in this field:

- A relatively static database environment which is investigated with dynamic queries. This is known as free-text search or document retrieval.
- A dynamic database environment which needs to be filtered with respect to relatively static queries. This is known as the filtering problem, current awareness or selective dissemination of information.

In a static database environment, the user formulates a query which is being matched against the documents in the database and the proper texts are returned to the user within seconds. A query in this context consists of keywords with optional wild cards. Its internal relations can be controlled by logical and statistical operators[22]. The data corresponding to a query can be retrieved very quickly, because the data collection has been pre-processed by generating an index over the database. An index contains all unique strings in the data collection, together with their positions in each document. Therefore, the index can be searched instead of the unordered database, which is virtually infinitely faster.

In a dynamic database environment, the user formulates a query or subscribes to an existing one, which corresponds to his or her personal interest. All incoming data is matched against these profiles and the proper texts are distributed to the user periodically. Although a query in this context is in principle syntactically identical to a query in a static database environment, this query's semantics is essentially different. In this context, a query's connotation resembles a user profile or interest description, which has a more enduring character. The incoming data must be indexed first before it can be matched and distributed according to the profiles. Once the index has been generated and stored in the database, together with the original data, the database environment becomes static for this passed period of time.

The user plays an active part in the retrieval process in a static database environment. In a dynamic database environment, the user formulates only once what he or she wants to be retrieved, for as long as the given profile corresponds with his or her interest.

There are some serious drawbacks in index-based information retrieval:

- For an average user, it often turns out be quite difficult to formulate a query which accurately corresponds to his or her intentions.
- Only documents which contain the query-keywords can be retrieved[23]. It has no ability to generalise over a query and cannot make incomplete matches well[24].

---

[22] Common logical operators are the conjunction (AND), the disjunction (OR) and the negation (NOT). An example of a statistical operator is the quorum (n of $\{k_1, k_2, ...\}$). This means that n keywords of the keyword-set must be in the document.

[23] To optimise retrieval, thesauri, or synonym-vocabularies, can be included in contemporary IR applications. However, this is not a real solution. The danger of retrieval-overkill increases significantly.

- No real context can be incorporated[25].

It would mean a large step for the field of IR to have a method which could incorporate these shortcomings without slowing down.

Artificial neural networks

Artificial neural networks are mathematical pattern recognition models which, although a variety of neural network architectures have been developed, all exhibit some interesting properties, important in an information retrieval context:

- Distributed data representation, i.e. x objects[26] are represented by y neurones. The ability to generalise over the data increases as the ratio between the number of objects, presented to the feature map, and the number of available neurones increases[27].

- Robust behaviour, i.e. the ability to process incomplete or incorrect information. This is a consequence of the distributed data representation and the ability to generalise over the data.

- Language independence, i.e. not the data itself is used during the process, but an internal vector representation, which is a series of numbers, representing a co-ordinate in the vector space.

From the large variety of neural network architectures, the Kohonen feature map has been implemented in the FILTER prototype.

---

[24] In contemporary IR applications a fuzzy search can be performed though. But since this fuzzy algorithm generates a number of keyword-permutations autonomously of the context, it can easily result in a retrieval-overkill.

[25] There do exist some context sensitivity-imitation techniques. An example is the binary proximity operator (*A* within *n* words of *B*), which searches for two keywords (*A* and *B*) within a range of *n* words. However, this is an incorporation of context in the statistical sense and not in the semantical sense.

[26] An object can, for example, be a natural language character or word.

[27] Architectures that do not have this property are not considered neural, but statistic table models.

Kohonen feature map

The Kohonen feature map is known to be an abstraction of the biological topology preserving maps found in the human visual system. It can be thought of as a two-dimensional grid. Each node in the grid contains a neurone, i.e. a set of input fibres or sensors or a vector representing a co-ordinate in the vector space. The data, which is trained to the feature map, is translated into vectors before it is presented. Therefore, the two-dimensional feature map can capture that portion of the multi-dimensional vector space which corresponds to the internal vector representation of the data.

The Kohonen formalism is a competitive learning algorithm. The two-dimensional feature map is a rectangular or hexagonal structure of neurones, which all have the same number of weights. The activation of a neurone, resulting from an input activation, is interpreted as a measure of correlation. The neurone, best representing the input activation, can therefore be determined by finding the neurone with the highest activity. In other words, the neurone, best representing the input vector, can be determined by finding the map vector with the minimum mathematical distance[28] with respect to the input vector. This neurone is called the best matching unit[29].

Once this neurone has been found, neurones within a certain region are adapted to some extent, depending on their distance from the best matching unit. Therefore, this region will recognise the current input better in the future. In time, the adaptation value and the region adaptation size also decrease to guarantee convergence. Because neighbouring neurones are updated with respect to an input vector's best matching unit each training cycle, a topological map emerges, holding related data elements in neighbouring regions[30].

To summarise, the feature map has some additional interesting properties, besides the general artificial neural network properties:

- Self-organisation on frequency and context, i.e. the frequencies of input patterns and overlaps between parts of these patterns, i.e. the patterns' context, are equally important.

---

[28] The most commonly used mathematical distance in vector space models is the Euclidean distance. This distance has also been used in this prototype.

[29] The common abbreviation for the Best Matching Unit is BMU.

Therefore, this automatic feature extraction out of unstructured data results in a map of conditional probabilities.

- Unsupervised training, i.e. the representation process of the training data in the feature map is fully automated. Therefore, one does not need to have any knowledge of the system architecture to be able to use such a system, if the system parameters are pre-configured.

- Topology preservation, i.e. if two object vectors are close to each other in the vector space, they will also be close to each other in the feature map after the training process. This results in a natural clustering of data features.

- Also, the Kohonen formalism is computationally efficient with respect to other neural architectures and it is relatively easy to implement.

Alternatives

Adaptive Resonance Theory (ART) [Carpenter *et al.*, 1991c] also encapsulates self-organisation and unsupervised training in a more neurobiologically founded manner. By integrating two subsystems, of which the higher-level subsystem supervises the lower-level subsystem, a stable and dynamic neural environment can be created. However, the working becomes quite complex, due to the many parameters involved. Also, the algorithm is computationally expensive.

The Simple Recurrent Network (SRN) [Elman, 1990] uses a recurrent network, where the hidden layer units are fed back into the input layer. Training such a network can also be considered unsupervised. By using recurrent input fibres, the model implements a higher order Markov chain[31]. Therefore, the network will contain a too specific representation of the data after the training process. Another known problem is the long training time, making it computationally expensive.

Other common neural architectures lack self-organisation and unsupervised training. These properties are important in the filtering problem, since it is not known in advance what ought

[31] A Markov chain is a mathematical model for event prediction. It was developed by A.A. Markov. The method is used to predict the possibility of an occurrence, given a history of occurrences. In general, in a high order Markov chain, an event can be a complex function. In the case of natural language data, the order of the Markov chain represents the number of preceding characters which are used to predict the next character.

to be trained. For this reason the Kohonen feature map was implemented for the FILTER prototype.

Neural filter

The neural filter algorithm implements a mechanism in which a query or user interest or profile, stated in natural language, is taught to a self-organising neural network, which derives an internal representation of the text. This representation is then matched against a continuous stream of incoming, unstructured data. The general set-up can be seen in Figure 10.1 below. Optionally, multiple queries can be matched simultaneously.

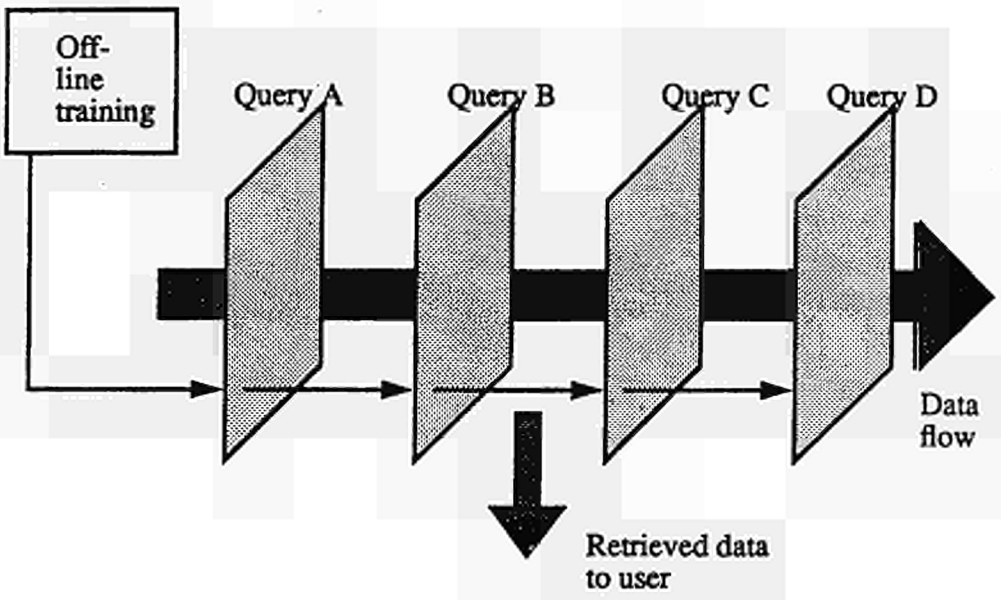


FIGURE 10.1: PRINCIPLE OF THE NEURAL FILTER (REPRINTED FROM [SCHOLTES 1993]).

Several algorithmic variants are possible, depending on the choice of objects which are presented to the neural network. In this project, characters have been used as basic objects to automatically incorporate context and maintain a more direct language independence[32]. This

[32] If words would have been used as objects, a dictionary would have been necessary. Such a dictionary could have been generated in advance though, by a statistical frequency algorithm. However, in the FILTER prototype language-dependent noise-

variant is known as the *n*-gram analysis method (see Figure 10.2) A *n*-gram is a *n*-length sequence of characters[33]. The *n*-gram analysis method can be interpreted as a window of size *n*, which is being shifted over the text. It is implemented in the Kohonen input sensors by assigning several sensors to each object within the window and concatenating all the window sensors to one big input vector. By shifting this window over the training text, only frequent *n*-grams form clusters on the feature map, infrequent patterns are overruled.

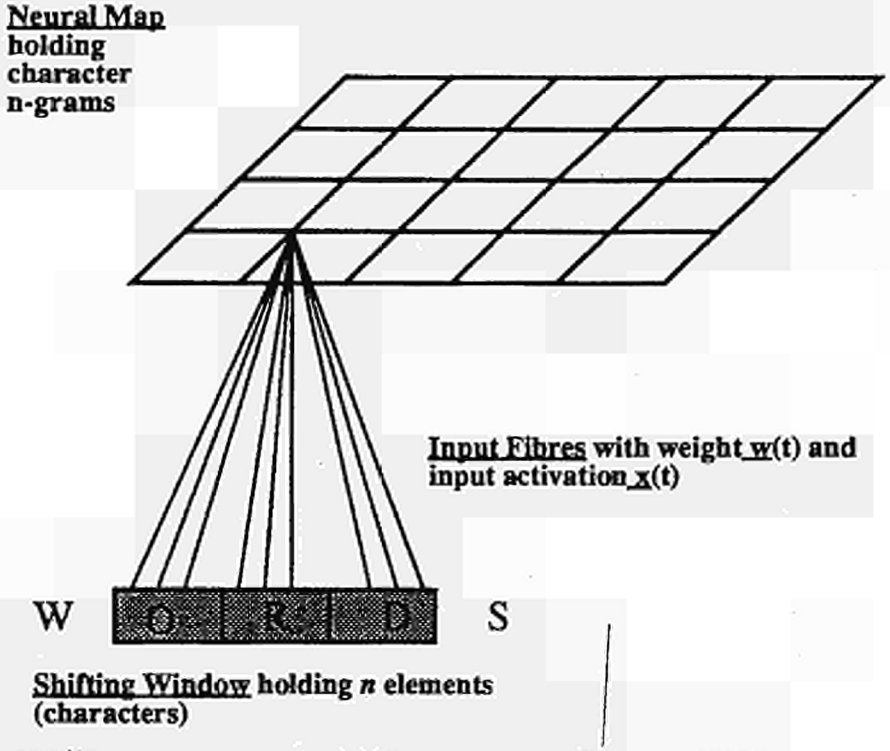


FIGURE 10.2: N-GRAM ANALYSIS METHOD (REPRINTED FROM [SCHOLTES 1993]).

After training, the input values of texts, mediated through the shifting window, which correspond to the query representation in the feature map, will yield low normalised cumulative errors and a high number of normalised cumulative perfect hits. This means that there is a certain degree of resemblance, or correlation, between these two texts. Therefore, if the feature map is used this way, it can be incorporated as a filtering device in an environment with relatively static queries and a dynamic information flow. This approach can also resolve some of the drawbacks of index-based retrieval.

---

words and noise-word endings can be eliminated to optimize performance. But, these noise-lists could also be generated in advance by a statistical frequency algorithm.

[33] For example, the set of possible trigrams, i.e. *n*=3, with the word TRIGRAM is : ??T, ?TR, TRI, RIG, IGR, GRA, RAM, AM?, M??}, where ?'s indicate variable context characters.

The schematic version of the algorithm is given in Table 10.1 below:

- **Initialise objects**
- **Initialise feature map**
- **Initialise input sensor**
- **Initialise text part statistics**
- **Teach query to feature map**
    - Filter data in chunks
    - Eliminate non-alphabetic characters and separate all words with a space character
    - Convert lower case characters to upper case
    - Optionally eliminate non-relevant words, non-relevant word endings and space characters
    - Shift window over filtered data to determine n-gram patterns
    - Convert patterns to vectors and copy in the input sensor
    - Present input sensor to feature map
    - Determine BMU
    - Determine current map region size to be updated
    - Determine current learn rate
    - Adjust the region of the BMU
- **Extract text parts from data flow**
    - Filter data in chunks
    - Eliminate non-alphabetic characters and separate all words with a space character
    - Convert lower case characters to upper case
    - Optionally eliminate non-relevant words, non-relevant word endings and space characters
    - Shift window over filtered data to determine n-gram patterns
    - Convert patterns to vectors and copy in the input sensor
    - Present input sensor to feature map
    - Determine the error of the BMU
    - Update text part statistics
    - Determine correlation between query and text part, based on the text part statistics

TABLE 10.1: SCHEMATIC VERSION OF THE NEURAL FILTER ALGORITHM.

10.2 Prototype

In this section we discuss the application properties on which we focused during prototype development: flexibility, performance, visualisations of the feature map and the accessibility issue.

An overview of the prototype in the form of an imaginary session will be given as well.

Properties

During the prototype development, four application properties were considered essential to create a properly functioning neural filtering environment:

- Flexibility, i.e. the ability to adjust and execute any valid event at any time to render interactive research.
- Performance, i.e. the speed at which accurate retrieval can be achieved.
- Visualisation, i.e. the clarification of the processes and the data by viewing these from different perspectives.
- Accessibility, i.e. the storage and retrieval of all input and output to enable reconstructions and variations.

Flexibility

Flexibility is of importance to interactive research. One has to be able to efficiently experiment with the process parameters.

An event-driven, multitasking environment is needed to provide maximum user-interactivity. This implies an Object Oriented Programming (OOP) concept.

In FILTER, every event can be fine-tuned, or even redefined, at *any* point in *any* process by a set of parameters and preferences.

Performance

Performance stresses the importance of execution speed in this type of applications. The importance of the accuracy of the retrieval is merely implicitly accentuated here, because this has been considered an obvious goal to achieve. However, if accurate retrieval cannot be achieved at a high speed, the system will simply not keep up with the incoming data flow in a

real-time filtering situation. Then, the system would still be useless, regardless of its retrieval accuracy.

The filtering process consists basically out of four continuously repeated events:

- Read the incoming data.
- Convert the data to patterns.
- Convert each pattern to its vector representation.
- *Search* the nearest neighbour in the feature map for each input vector.

The most time-consuming event in a single processor[34] environment is the nearest-neighbour search, because for each input vector the whole feature map must be searched. Therefore, two possible optimisations have been investigated to speed up this event.

Dynamic k-d tree

A k-d tree is a tree structure for storing and retrieving k-dimensional data points. Although the k-d tree is usually being built during training, it should also be possible to convert the feature map into this structure after training by dividing the feature map recursively into two equal neurone collections along the axis of greatest range. The data points are stored in the leaf nodes. By using this search technique, the search time decreases exponentially[35], if the tree is in balance. See also Figure 10.3.

---

[34] The conceptually most obvious minimisation of execution time, i.e. the implementation in a multiprocessor environment, has been ignored in this report for practical reasons.

[35] The full-search algorithm has a complexity O(N), where N is the number of neurones in the feature map. The tree-search algorithm has a complexity O(log N).

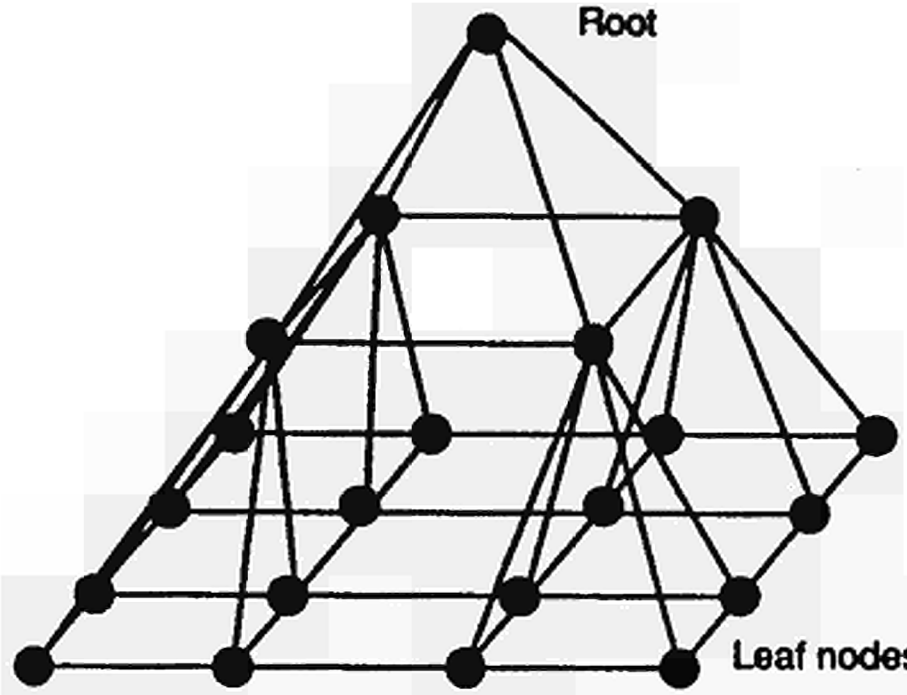


FIGURE 10.3: FEATURE MAP ELEMENTS AS LEAF NODES IN A TREE STRUCTURE (REPRINTED FROM [KOIKKALAINEN 1990]).

In the implementation, each internal node contained an average vector of the data co-ordinates of its two daughters. Unfortunately though, it turned out that these average vectors levelled too much after a few tree levels. Relatively often, this led to inaccurate retrieval of a pattern's best matching unit.

Although the k-d tree structure could be an efficient representation of the feature map, the nearest-neighbour search would still dominate the filtering process as the most time-consuming event. The distance between each input vector and some map vectors must still be calculated. Therefore, another approach called the *Table map* was tried.

Table map

Hashing is a well known statistical addressing technique which retrieves the output by calculating a function of the input. In this case, this means that the distance of the best matching unit must be returned, based on the input patterns. To accomplish this, all possible patterns must be generated, each pattern must be matched against the feature map and each distance must be stored at the position in the hashing object which represents its pattern. This can take some time, but it has to be done only once for a trained feature map. This way, the actual filtering process events can be replaced by:

- Converting each pattern to its hashing address.
- *Get* the distance, contained in that address.

However, depending on the context size, there can be very many possible patterns[36]. To enable storage capacity for up to a hundred million distances, a two-dimensional hash table was implemented. In the prototype, this is called a table map, analogous to the feature map and the vector map. In practice though, a table map with a hundred million entries is not efficient anymore. It takes a lot of memory and preparation time[37]. Therefore, if the table map is to be used, the context size should be kept low. In the evaluation section, it will be investigated whether a low context size is possible or not, in relation to the retrieval accuracy.

Visualisation

In dealing with visualisation, the importance of viewing the data from different perspectives is stressed, in order to help us clarify what exactly is happening with the application objects.

Two types of data visualisation have been implemented:

- Textual visualisation, i.e. a print of the contents of an object in ASCII-format. This can be useful as a low-level clarification source.

---

[36] The number of possible patterns is O^c, where O is the number of characters in the language and c is the context size.

[37] The amount of memory needed for each table, containing 10000 Euclidean distances, is 80 Kb. This means that if the number of entries is $27^3 = 19683$, the table map will take 160 Kb of memory. (Generation will take about 8 minutes, depending on the feature map size.) If the number of entries is $27^4 = 531441$, the table map will already consume 4.3 Mb of available memory. Preparation time here includes generation, saving and loading time.

- Graphical visualisation, i.e. a print of relations within an object in a unrestricted format. This is very useful as a higher-level clarification source.

Both textual and graphical visualisation can reflect two different perspectives on an object:

- Static perspective, i.e. a print is a snapshot of an object.
- Dynamic perspective, i.e. the print is an anchored view onto an object to follow the process.

Textual visualisation

Among the static textual visualisations are the feature map which can be printed to view the neural weights and the vector map which be printed to clarify the internal vector representation of the data. Also, the contents of the error- and activity recording objects can be printed to enable customised visualisation with an external spreadsheet application.

Dynamic textual visualisation has been applied to the internal data flow of both the query and the passing data to examine how exactly the input is transformed, before it is presented to the feature map.

Below is a fragment of the textual visualisation of the contents of a feature map. Each neurone in the feature map consists of ((Sensors per object)*(Context size)) weights. These concatenations of weights are used as vectors, representing co-ordinates in a multi-dimensional space. This example shows a cluster in the fifth and sixth dimension of several neurones, which means that this region will recognise patterns, which have a character in the middle which maps to 1.000000 (see also the vector map in Table 10.3).

```
FEATURE MAP initialised with current settings...
        X-dimension                 : 13
        Y-dimension                 : 17
        Context size                : 3
        Sensors/Neurone             : 9
        Random spread               : 15
neurone[0,0]    : 0.224287 0.952296 0.029184 0.029922 0.687756 0.486086 0.873835 0.900448 0.889078
neurone[0,1]    : 0.127397 0.976929 0.028571 0.372500 0.536198 0.871545 0.803125 0.985772 0.996528
neurone[0,2]    : 0.303389 0.937831 0.002000 0.496275 0.509724 0.998167 0.492377 0.996237 0.997918
neurone[0,3]    : 0.377346 0.997539 0.003503 0.542813 0.995419 0.999993 0.058903 0.779591 0.805812
neurone[0,4]    : 0.697131 0.958209 0.025608 0.967122 0.999992 0.999998 0.091885 0.198423 0.993085
neurone[0,5]    : 0.558126 0.403755 0.226493 0.997551 0.999998 0.999998 0.006682 0.016472 0.949989
neurone[0,6]    : 0.974968 0.136078 0.503579 0.994403 0.999998 0.999998 0.125672 0.570563 0.768181
neurone[0,7]    : 0.697058 0.043202 0.876233 0.807937 0.999998 0.999998 0.040433 0.202237 0.363312
neurone[0,8]    : 0.343057 0.355185 0.932372 0.756739 0.999777 0.999998 0.808857 0.021074 0.368147
neurone[0,9]    : 0.096726 0.503642 0.512291 0.715875 0.999692 0.999998 0.993122 0.076557 0.223334
neurone[0,10]   : 0.597828 0.829262 0.429358 0.926244 0.999499 0.999998 0.791958 0.497056 0.106886
neurone[0,11]   : 0.502821 0.997213 0.978075 0.999931 0.999998 0.999998 0.449903 0.631282 0.184550
neurone[0,12]   : 0.685060 0.869337 0.995705 0.894068 0.902305 0.999998 0.661747 0.613795 0.036563
neurone[0,13]   : 0.798250 0.979789 0.998584 0.509939 0.576002 0.999993 0.676429 0.866105 0.002862
neurone[0,14]   : 0.922398 0.594168 0.717885 0.372101 0.523254 0.831094 0.591289 0.945196 0.592083
neurone[0,15]   : 0.988563 0.380237 0.419349 0.022776 0.403899 0.420415 0.513318 0.999918 0.999917
neurone[0,16]   : 0.901105 0.286195 0.012265 0.066698 0.544840 0.558795 0.600750 0.999992 0.999998
neurone[1,0]    : 0.242177 0.625855 0.311376 0.016595 0.297078 0.142991 0.847766 0.988551 0.100470
neurone[1,1]    : 0.084173 0.956070 0.631514 0.031282 0.485015 0.505438 0.518370 0.998363 0.836606
neurone[1,2]    : 0.295684 0.815231 0.462251 0.262446 0.609304 0.773652 0.245971 0.922429 0.826050
```

TABLE 10.2: FEATURE MAP: TEXTUAL VISUALISATION.

The vector map below defines the mapping for each data pattern into its internal vector representation. The example is from an artificial data set reflecting the order of the alphabet.

```
VECTOR MAP initialised with current settings...
        X-dimension         : 27
        Y-dimension         : 3
        Code spread         : 2
A       : 0.000000 0.000000 0.000000
B       : 0.000000 0.000000 0.500000
C       : 0.000000 0.000000 1.000000
D       : 0.000000 0.500000 0.000000
E       : 0.000000 0.500000 0.500000
F       : 0.000000 0.500000 1.000000
G       : 0.000000 1.000000 0.000000
H       : 0.000000 1.000000 0.500000
I       : 0.000000 1.000000 1.000000
J       : 0.500000 0.000000 0.000000
K       : 0.500000 0.000000 0.500000
L       : 0.500000 0.000000 1.000000
M       : 0.500000 0.500000 0.000000
N       : 0.500000 0.500000 0.500000
O       : 0.500000 0.500000 1.000000
P       : 0.500000 1.000000 0.000000
Q       : 0.500000 1.000000 0.500000
R       : 0.500000 1.000000 1.000000
S       : 1.000000 0.000000 0.000000
T       : 1.000000 0.000000 0.500000
U       : 1.000000 0.000000 1.000000
V       : 1.000000 0.500000 0.000000
W       : 1.000000 0.500000 0.500000
X       : 1.000000 0.500000 1.000000
Y       : 1.000000 1.000000 0.000000
Z       : 1.000000 1.000000 0.500000
_       : 1.000000 1.000000 1.000000
```

TABLE 10.3: VECTOR MAP: TEXTUAL VISUALISATION.

When the user requests a snapshot during a process, an anchored view is connected to this process. As long as the user does not explicitly request disconnection, the evolving relations are shown. A request for a snapshot after a process can be considered as a visualisation of the final state in that process. During and after the training process, the state of self-organisation for each pair of dimensions can be examined. Here, the relations between neighbouring neurones are visualised as co-ordinates in the vector space. Also, the interneuronal distances can be visualised as can be seen in the following figure.

The fragment in Figure 10.4 shows the feature map from another, more static perspective, where cluster boundaries can easily be traced. The Euclidean distances between neighbouring neurones are presented as thickness degrees in a static grid.

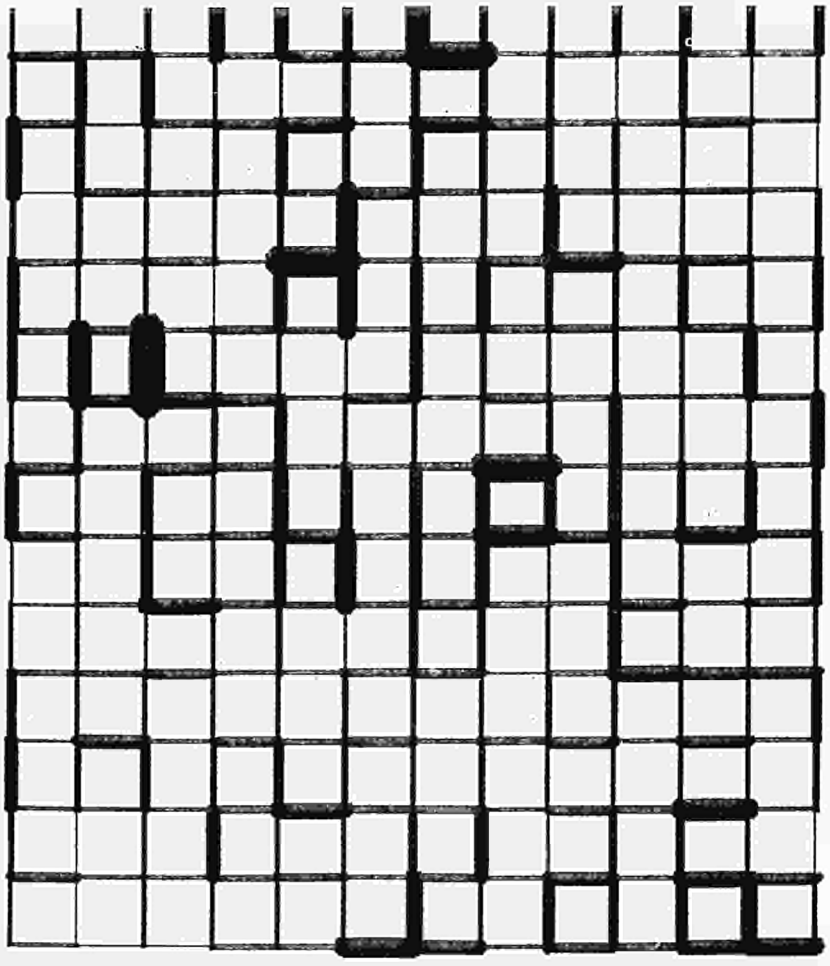


FIGURE 10.4: INTERNEURONAL DISTANCES VISUALISATION.

Another perspective is offered by the object distribution. The objects, or n-grams, can be examined from two perspectives. The first perspective visualises the best objects possible for each neurone (see Figure 10.5). The visualisation of the distribution of the best objects in the query requires a statistical frequency analysis[38] in the pre-processing phase. This results in an ideal object distribution, which can be useful in a comparison with the actual object distribution. To optimise comparisons, the actual object distribution visualisation includes a search mechanism where any object can be entered and its best matching unit is returned.

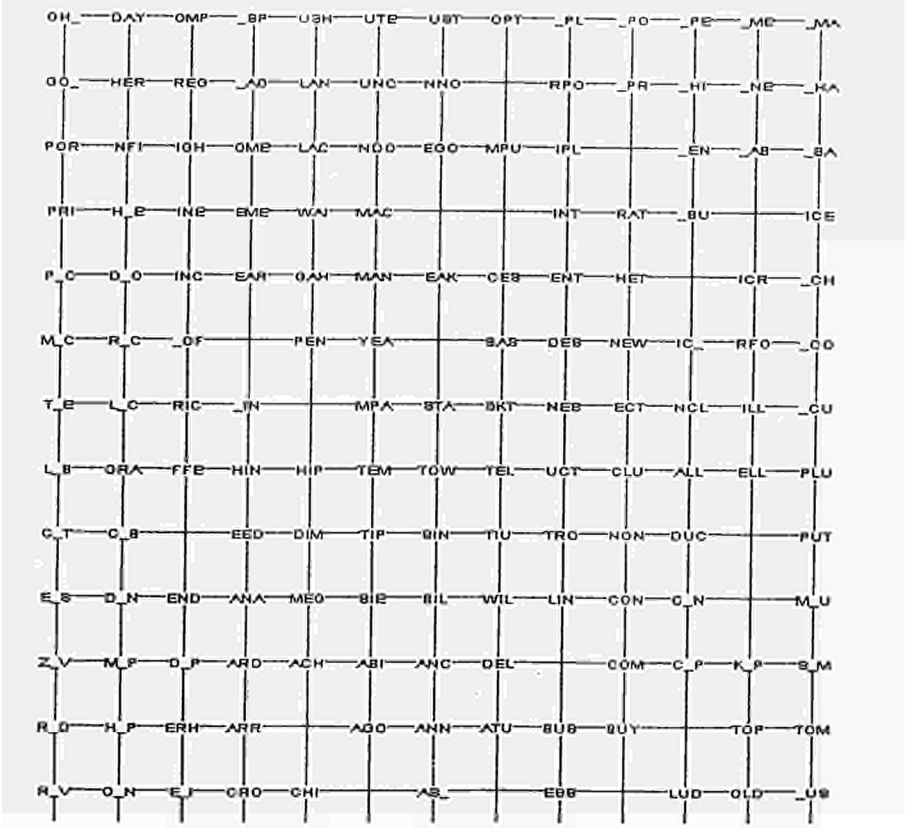


FIGURE 10.5: OBJECT DISTRIBUTION (BEST OBJECTS IN QUERY - INCOMPLETE).

The empty co-ordinates are due to the fact that some patterns have the same neurone as their BMU. Only the most frequent object is printed in that case.

---

[38] In the implementation, a dynamic B-tree has been used to minimize execution time. The table map could not be used, because of its inefficient nature in the case a high context size is used. The B-tree object has also been applied to the common words to maximise the overall performance of the prototype.

Long term development of the map can be examined by visualising the error recording (see Figure 10.6). This object contains the Euclidean distances for each best matching unit for each pattern in the query and is visualised in a graph. The activity recording (see Figure 10.7) contains the complements of the Euclidean distances for each best matching unit for the 500 most recent patterns in the passing data. It is visualised in a graph, in relation to the hit threshold as well as the view threshold.

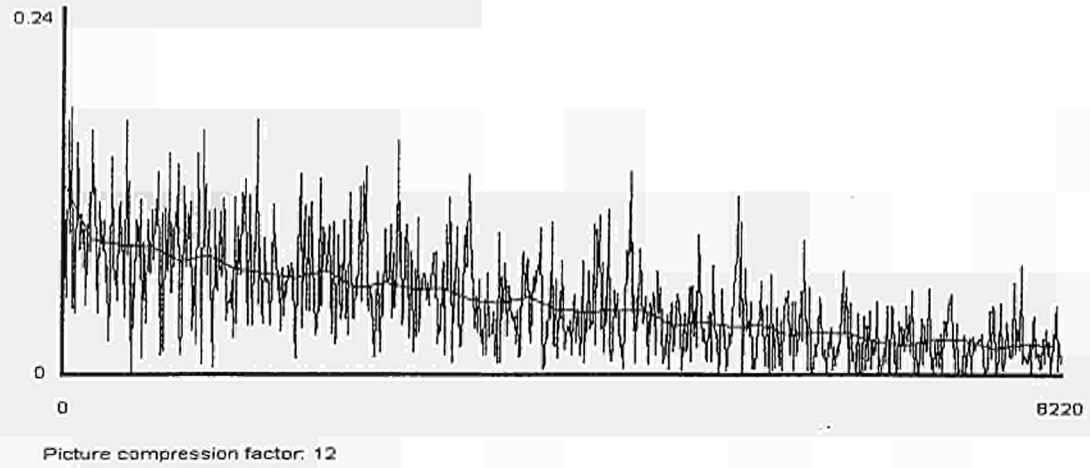


FIGURE 10.6: ERROR RECORDING.

The X-axis represents the maximum range of the pattern's Euclidean distances to their BMU's during the training process. The Y-axis represents the number of training cycles. In the evaluation, the queries have been presented ten times to the feature map. As the feature map converges to its final representation, the overall errors get smaller. This means that the feature map finds an overall way to represent the query. Peeks are due to infrequent patterns. Picture compression means that only one per twelve errors is plotted to give a overall impression.

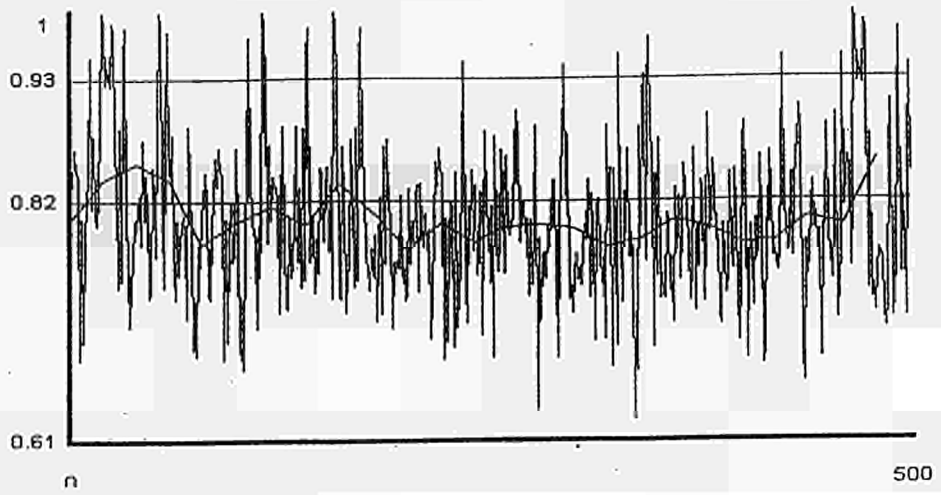


FIGURE 10.7: ACTIVITY RECORDING.

The X-axis represents the maximum range of the complement of the pattern's Euclidean distances to their BMU's during the extraction process. The Y-axis represents the most recent extraction cycles. Here, the upper boundary (at 0.93) represents the perfect hit threshold. The lower boundary (at 0.82) represents the view threshold.

Accessibility

The concept of accessibility stresses the importance of storage and retrieval of all input and output to enable reconstructions and variations of experiments. Therefore, the prototype supports three document-view formats, which are invoked by a open/save-command:

- FILTER Output (*.out / *.txt), i.e. the standard ASCII text format, used for textual visualisations, the system report and external data.
- FILTER Picture (*.pic), i.e. a special format which supports system- and user drawing, as well as dynamically sizeable text. This format is used for graphical visualisations.
- FILTER Hitlist (*.hit), i.e. a special database format which uses Open DataBase Connectivity (ODBC) to connect only to external DBase (*.dbf) databases with the prototype database structure.

The prototype also supports four document-object formats, which are invoked by a load/save-command:

- FILTER Settings (*.set), i.e. a special, protected format which contains the configuration of parameters, preferences, files and paths and system settings.
- FILTER Demo (*.dem), i.e. a derivation of the settings format which also contains the special demo settings, which causes the demo mechanism to use artificial vectors instead of natural language data. This is useful for simulations of ideal situations.
- FILTER Data (*.dat), i.e. a special ASCII format which contains the contents of all objects, together with their dimensions, except the table map (because of its optional nature).
- FILTER Table (*.tbl) , i.e. a special ASCII format which contains the table map with its dimensions.

With these formats, the prototype does provide maximum accessibility. Especially the FILTER Hitlist format is of great importance, because it enables each user to optionally decide the amount of returned information by merely resetting the view threshold.

A Session with the FILTER Prototype

To conclude this section, a global overview of the prototype will be given in the form of an imaginary session.

When the session is started, a workspace appears, containing eight menus, a toolbar, a status bar and a system log-file. This log-file reports that the most recently used settings file has already been reloaded.

If this session is to be a continuation of the last session, only the corresponding data has to be loaded. After loading the data, all menu commands become accessible, indicating that the pre-processing phase, i.e. all processing needed to start the extraction process, has been completed.

However, if this session is not to be a continuation of the last session or if there is no data file which corresponds with these settings, the complete process cycle has to be pursued. To start with, the settings which correspond best to the current aim should be loaded. To alter these settings, the Parameters dialogue, the Preferences dialogue and the Files and Paths dialogue can be opened to reconfigure these settings. Within the Parameters dialogue, the Hints dialogue can also be opened, which can be of great help to optimise the settings. Having fine-tuned this configuration, these new settings ought to be saved.

Now the system has been configured, the actual process can be started. First, the data objects have to be initialised according to these settings. Next, the query must be taught to the feature map. Depending on what visualisation preferences have been set, a number of new windows appear, which offer views from different perspectives onto the training process. These visualisation preferences, like all settings, can be altered at any moment. For example, if the representation process does not seem to work out well, it is possible to interfere by adjusting training parameters like epsilon, i.e. the learn rate, and sigma, i.e. the region update area. Of course, the process can also be interrupted.

When the training process ends, the system asks whether a table map for this feature map must be generated. If a low context size has been used, this question should be answered affirmative. After the table map generation, saving the data is necessary to be able to reuse these data objects in future sessions. The system asks whether the table map should also be saved.

At this point, the pre-processing phase has been concluded. All commands are accessible now, including the Extract Text Parts command. When the extract process is started, three

more new windows appear: the internal data flow view, the feature map activity view and the hitlist view. The first two views can be inactive, depending on the preferences settings. The hitlist view cannot be deactivated. When this view has the focus, the hitlist-navigation toolbar buttons become activated to provide a convenient way to also browse through the database as the hitlist is being build. When the contents preview, contained within the hitlist view, looks interesting, the Retrieve button in the view can be pushed to examine the whole text part in a separate view. The hitlist can also be ordered on one of the five available fields during this process. To fine-tune retrieval, the view threshold can be adjusted to increase or decrease the number of retrieved documents. This can best be done after examining the activity view, because it shows the activity in relation to the thresholds. To activate a new view threshold, the sort hitlist-command must be called to update the database.

Once the extraction process has ended, the hitlist ought to be saved to enable future access. Now the hitlist can be printed, reordered, edited and reviewed at all times.

The imaginary session described here, assumes the user is willing to experiment somewhat within this neural filtering environment. If this is not the case however, the process cycle can be minimised with respect to the user effort by activating the evaluation mode.

In this evaluation mode, all the user has to do is prepare a number of settings, select those settings in the Evaluation dialogue and press the Evaluate button. This mode implicitly saves all data and hitlists. In this mode, the user can simply do something else on the computer, because the prototype also supports smooth background processing.

10.3 Evaluation

This section explains how the prototype, as described in the previous section, has been evaluated. First it describes the preparation phase, i.e. how the data set was composed, how the queries were selected, how the settings, or the parameter configurations, were set. Then it is described how the correlation, or the degree of resemblance, between the query representation in the feature map and each document in the data set was calculated.

After that we describe the execution & analysis phase, starting with a detailed report on the preliminary outcomes. Based on these preliminary outcomes, additional tests and analyses have been carried out, which are also described in detail.

After that a comparison is made between the FILTER prototype and ZyIMAGE, a contemporary index-based information retrieval system.

Preparation

Data set

The data set consisted of 100 image-based, rather accurately scanned, articles (823 KB) from *the Wall Street Journal Europe, August 8-18, 1994.* The article collection was composed by querying ZyIMAGE, an index and image based document retrieval application. Three elementary queries or interest profiles were used: WAR*, EC and COMPUT* . The inclusion of wild cards ensured that the recall would be high and the precision low . In theory, there should have been three topics in this data set, one for each query. In practice however, there was only one coherent group: the COMPUT*-group. This is partly because this group was composed out of documents with a relatively high hit density only. Another factor could be that words beginning with the substring WAR have too diverse semantics, whereas the string EC is too specific. Words beginning with the substring COMPUT all seem to belong to the same semantic class "Computer terms". The composition of the data sets for the three interest profiles can be seen in Tables 10.4 a,b,c,d.

TABLE 10.4 A: DATA SET COMPOSITION. QUERY 1: WAR* (NO FUZZY SEARCH WAS USED) COMPLETE RETRIEVAL, PRESENTED IN DESCENDING HIT DENSITY:

Reference	Document	Hits	Keywords
W1	QJ.TXT	15	Warburg, investment bank, hostile takeover
W2	FZ.TXT	2	federation of taxpayers, warns, tax burden taxpayers
W3*	13V.TXT	7	warnings, Compaq, computer keyboards, wrist injuries
W4	G5.TXT	4	Time Warner inc, Viacom inc. sells theater chain
W5	1F6.TXT	2	fare war, eurotunnel denies fare reductions
W6	10I.TXT	23	marketing & media, iced tea to europe, warner bros. records
W7	10G.TXT	6	Unilever, Omo power, detergent war

W8	AW.TXT	6	Cisco systems, communicating with small investors
W9	X8.TXT	4	pharmaceutical industry, takeover, Glaxo holdings
W10*	13W.TXT	2	IBM, order system for software
W11	18K.TXT	1	Alliance Pharmaceutical, Johnson & Johnson, drugs group
W12	10H.TXT	4	Unilever, Omo power, Proctor & Gamble, soap war
W13	PX.TXT	3	Unilever pretax profit rose
W14	13O.TXT	1	wholesale prices fall in western Germany
W15	JM.TXT	4	car trip in Europe, reasons to stay at home
W16	AJ.TXT	6	business opportunity in Poland, financing
W17	AK.TXT	1	Russia plans to take action on plague of unpaid bills
W18	9T.TXT	5	Murdoch's price war, newspaper sales up, profits are falling
W19	Q7.TXT	3	World Bank warned Turkey, international loans and credits
W20	G2.TXT	2	Polish court, license, PolSat TV, nationwide broadcasting
W21	X3.TXT	2	U.S. industrial production +0.2% in July, warm weather
W22	70.TXT	1	Britain, Smart-card technology, drivers, EC directive
W23	9X.TXT	3	British monetary-policy, industrial production, Warburg
W24*	1HI.TXT	2	Hewlett-Packard share price rises on increases in earnings
W25	Q1.TXT	2	microwave filter, air warfare, communications applications
W26	10K.TXT	2	M6, French television, bourse listing
W27	FP.TXT	3	German teens find new fuel for disco raves, Red Bull
W28	FS.TXT	1	Kuwait seals Russian ties with major arms purchase
W29	AZ.TXT	2	China's elusive effect on market for commodities
W30	AE.TXT	2	U.S. FCC, license, interactive television services
W31	D0.TXT	1	ING Group, Bank Brussels Lambert, takeover
W32	CL.TXT	2	British Airways, rise in pretax profit, shares fall
W33	CK.TXT	1	Lending in U.K. to consumers rises to record
W34	AL.TXT	2	AIDS, epidemic, research, conference
W35	6K.TXT	2	CNN, BCC, Cox, 24-hour news business, competitors
W36	PM.TXT	2	NATO begins search for new top secretary-general
W37	PS.TXT	1	TBB, CAA, pact, video programming, telephone customers
W38	73.TXT	2	German postal service will offer 25% of shares to public
W39	9S.TXT	2	Europe's economic recovery is beating expectations
W40	JN.TXT	2	Zurich, Fust Investors, takeover, insider trading
W41	71.TXT	1	costly diet products in Russia, Herbalife, scientific?
W42	CR.TXT	1	Saatchi & Saatchi Co., profit, British advertising, marketing
W43	1HD.TXT	3	American Express, data mining, Sybase Inc
W44*	A0.TXT	1	IBM's overhaul of disk-drive unit may cut jobs in Europe
W45	AQ.TXT	1	Union Bank of Switzerland, shares drop 28%
W46	A1.TXT	1	Finalists to purchase Kodak's household-products division
W47	CX.TXT	3	China should take a hard line on software pirates
W48	CH.TXT	1	Italian Silvio Berlusconi, television advertising, RAI
W49	GD.TXT	1	German teens find new fuel for discos, continued
W50	XL.TXT	2	Toys R Us, investors, Petrie Stores, deal, Warburg complex
W51	146.TXT	2	new markets in central and eastern europe, investors
W52	CQ.TXT	1	Germany, Bayer AG, buying drug business of Kodak
W53	6Y.TXT	1	global news business, BBC challenges CNN
W54	AR.TXT	1	European central banks, Germany's Bundesbank, discount
W55	1JN.TXT	2	Russian mafia, new problems, La Cosa Nostra
W56	13P.TXT	1	American Home Products buys American Cyanamid
W57	10B.TXT	1	German drug investigation, two Schering AG drugs
W58	X2.TXT	1	Portuguese Bank at war, free-market policies
W59	QA.TXT	1	Algeria, nationalist guerrillas, Islamic slogans, war
W60	WZ.TXT	1	Carlos the Jackal, arrest, terrorism, politics, Sudan, France
W61	FQ.TXT	1	French cognac, consumption dropped, Norwegians buy cars
W62	X1.TXT	1	trends, business travel and tourism, economic recovery
W63	PL.TXT	1	Maastricht, fiscal policy, Europe, monetary union
W64	6H.TXT	1	scandals, Washington, Congress, lobbyist-paid trips
W65	1JU.TXT	1	LDDS Communications inc., telephone, acquisition, WilTel
W66	10A.TXT	1	Russia, Germany, plutonium smuggling, pressure
W67*	CG.TXT	2	Reebok, workers' rights, China, software, aquarium, fish
W68	6S.TXT	1	Ted Turner, New Line Cinema, Hollywood, movie business
W69**	A3.TXT	1	Dell Computers, return, notebook, new designs, price battle
W70	13S.TXT	1	Scandinavian Airlines System, recovery, pretax profit
W71	G4.TXT	1	record stores, changing, multimedia stores
W72	G7.TXT	1	cars, Renault SA, front-runner in French privatization race

W73	1HG.TXT	1	U.S. retailing, Dayton Hudson, Wal-Mart Stores, earnings
W74	1JO.TXT	1	Johnson & Johnson acquires Neutrogena, personal care
W75	1IG.TXT	1	U.S. health care, problem,subsidized by federal government
W76	13G.TXT	1	Vietnam places hope for economic health in local enterprise
W77	6O.TXT	1	American Cyanamid, American Home Products, takeover
W78	JK.TXT	1	Japanese price revolution, consumer behaviour, shopping
W79	13U.TXT	1	technology & health, employees, virtual offices, low morale
W80	GA.TXT	2	Boston sees payoff and problems in east Europe, expanding
W81	B1.TXT	1	Asian markets, bourses consolidate after gains, Tokyo
W82	PJ.TXT	1	Bill Clinton, defeat on Crime Bill, Washington, Congress
W83	1HU.TXT	1	Helsinki, Oy Nokia, toilet-paper, cellular phones
W84	G3.TXT	1	British, high taxes, bargain hunters dent U.K. alcohol sales
W85	B0.TXT	1	Eurobond Market, quiet week
W86	GK.TXT	2	international bond indexes, bund prices fall

TABLE 10.4 B: DATA SET COMPOSITION. QUERY 2: EC (NO FUZZY SEARCH WAS USED) COMPLETE RETRIEVAL, PRESENTED IN DESCENDING HIT DENSITY:

Reference	Document	Hits	Keywords
E1	13J.TXT	1	EC lobbyists, American Express, Brussels
E2	FY.TXT	1	Terra Industries buys fertilizer products concern (NO ec!)
E3	10I.TXT	1	marketing & media, iced tea to europe, warner bros. records

TABLE 10.4 C: DATA SET COMPOSITION. QUERY 3: COMPUT * (NO FUZZY SEARCH WAS USED). INCOMPLETE RETRIEVAL (THE 18 HIGHEST RANKED DOCUMENTS OUT OF 50 ARE SELECTED), PRESENTED IN DESCENDING HIT DENSITY:

Reference	Document	Hits	Keywords
C1**	1F7.TXT	6	Dell Computers, overhaul, desktop computers,Pentium,Intel
C2**	A4.TXT	7	Compaq's flagship line of notebook computers, defect
C3*	13V.TXT	10	warnings, Compaq, computer keyboards, wrist injuries
C4*	13W.TXT	6	IBM, order system for software
C5**	1QF.TXT	9	IBM plans to slash prices to counter Compaq, overhaul of PC line
C6*	10F.TXT	5	bidding for Ziff Communications, computer magazines
C7*	XA.TXT	7	European PC sales gained, Compaq, IBM, Apple
C8**	A3.TXT	7	Dell Computers, return, notebook, new designs, price battle
C9	AD.TXT	1	Lufthansa AG, freight, maintenance, computer operations
C10	1QD.TXT	1	recognition factor, Swiss Bank, note-counting, manual labor
C11*	1QW.TXT	4	AT&T, Intel, software methods, PC-Based, videos
C12	108.TXT	1	remote answering-machine service, computer technology
C13	XJ.TXT	6	technophobia, human skills vs. information technology
C14*	1HI.TXT	3	Hewlett-Packard share price rises on increases in earnings
C15*	1QV.TXT	5	detente between Novell and Microsoft, product tuning
C16*	A0.TXT	3	IBM's overhaul of disk-drive unit may cut jobs in Europe
C17*	1F9.TXT	1	CompUSA Inc., struggling U.S. computer-superstore chain

TABLE 10.4 C: DATA SET COMPOSITION: NOTES

A number of documents have multiple occurrences

- 13V.TXT* : W3 ⇔ C3.
- 10I.TXT : W6 ⇔ E3.
- 13W.TXT*: W10 ⇔ C4.
- 1HI.TXT* : W24 ⇔ C14.
- A0.TXT* : W44 ⇔ C16.
- A3.TXT** : W69 ⇔ C8.

** means the document is closely related to the query C1. These 4 articles only are directly about computer manufacturers like Dell Computers and about one of their computer models. These articles must therefore be extracted.

* means the document is somewhat related to the query C1. These 10 articles are about computer-related companies. These articles may therefore be extracted.

Queries

Testing has been carried out with two types of queries:

- A *full-text query,* i.e. a document in natural language.
- An *artificial query,* i.e. a concatenation of keywords, separated by a non-character.

As the full-text query, the highest ranked document of the C-group (C1 in the above table) was chosen. By choosing a document from the data set as the query, the maximum activity for the neural net gets implicitly defined. This has been used to determine the relative activity of all other documents as well as to optimise the hit threshold. Taking the document with the highest hit density ensures that its dominant patterns, or features[39], are represented in the map after the training process.

The artificial query was composed by concatenating all informative words in the full-text query, separated by a comma. By deriving the artificial query from the full-text query a comparison between the results can be made. One advantage of the artificial query could be that, because of it is cleaned up from noise, a better representation can be formed on the feature map after the training process. A practical advantage is the smaller size of the neural net needed to represent the query, simply because the query is much smaller. This speeds up the overall performance of the system. Also a comparison with index-based information retrieval systems can be made, due to the artificial query's resemblance to a weighted quorum query.

The text of the full-text and the artificial query can be found in Table 10.5.

TABLE 10.5: FULL TEXT QUERY

C1 (1F7.TXT):

<scan_date> 8/22/94 </scan_date>
<source> Wall Street Journal Europe </source>
<title> NA </title>
<author> NA </author>
<copyrights> NA </copyrights>
<abstract> NA </abstract>
Dell Plans to Overhaul Desktop Computers Aimed at Companies
By a Staff Reporter
AUSTIN, Texas - Den Computer Corp. is expected to announce today an overhaul of its high-end OptiPlex line of corporate desktop computers that will include price cuts and the introduction of high-perform- ance Pentium microprocessors. The computer vendor said it would break,a price barrier by offering for under $3,000 a fully configured desktop system based on a speedy 90-megahertz version of Intel Corp.'s Pentium chip. "We are moving toward Pentium carry- ing over into the corporate side," said

[39] In the case of this query, in combination with trigrams, examples of features are COM, OMP, MPU, etc.

Doug MacGregor, Dell's vice president for desktop computers. "None of our corporate customers have a question of whether or not they'll use Pentium. It's a question of when." Dell has pushed hard with the newest Intel chip, and now more than half of all Dimension machines sold to home users and small businesses use Pentium chips, Mr. MacGregor said. But corporate customers have been waiting for Pentium prices to fall, he said. With the new Optiplex prices, businesses will be able to buy Pentium-based machines at prices similar to what they were paying less than a year ago for slower 486-based computers, Mr. MacGregor said. Dell said its new OptiPlex models replace machines introduced about one year ago. The new machines incorporate ad- vanced power management, enhanced net- working capabilities and easier-to-use "plug In' play" features.

<xref image="J:\INDEX\ALGEMEEN\TIFF\1F7_01.tif|0"> image: </xref>

TABLE 10.6: ARTIFICIAL QUERY

C1', derived from C1 (1F7.TXT):

Dell,Desktop,Computers,OptiPlex line,
desktop computers,introduction,high-performance,
Pentium microprocessors,computer,price barrier,
configured desktop system,speed,megahertz,
Intel,Pentium,chip,Pentium,Doug MacGregor,Dell,
desktop computers,Pentium,Dell,Intel chip,
Pentium chips,MacGregor,Pentium,Optiplex,
Pentium-based machines,486-based computers,
OptiPlex models,power management,networking,
"plug In' play"

Settings

Four parameters have been exhaustively tested:

- The generalisation factor, i.e. the ratio of the network dimensions to the number of n-grams in the query. Values of 2, 4 & 6 have been processed.
- The context size, i.e. the size of the window which is being shifted over the data. Values of 3, 5 & 7 have been processed.
- The space as character, i.e. the usage of the space character to include a word's natural context better in the representation in the feature map. Values of 0 & 1 have been processed. Note that this parameter will not be varied when the artificial query is processed. In that case there is *by definition* no natural context.
- The hit threshold, i.e. the degree of correlation a n-gram must have with the best matching neurone in the feature map to be recorded as a perfect hit. This is essential in the extraction process, if the hitlist is to be sorted on the perfect hit rate or the average hit error. Although this parameter could also be altered during or after the actual extraction process, this would of course make the perfect hit rate and the average hit error not reliable anymore. Therefore, in the preliminary evaluation, the hit threshold has been kept

constant at a value of 0.2. In the additional evaluation the hit threshold will be varied, based on the preliminary results.

Note that this means there has not been looked in detail into the form of the feature map, the learning rate and the weights-update region size during the training process, possible optimisations of vector-character assignments, and so on. All these parameters have been kept constant on values, based on experimental pre-processing as well as on former research by others.

Definitions of Correlation

Three levels of *correlation* were determined manually for all documents in the data set: extreme correlation, significant correlation and no correlation. Four documents **had to be** retrieved to achieve a maximum recall. Only these documents are directly about computer manufacturers *and* about one of their computer models. Ten other documents were **allowed to be** retrieved to continue a maximum precision. These documents are about computer-related companies or about computer-related products. All other 86 documents were rated irrelevant (see also the notes below the tables with the data sets)

The term information value has been used as a reversed value of a pattern's probability of occurrence. This means that if the probability a pattern will occur is high, its information value is low.

The term precision has been used to indicate the number of correlating documents which were retrieved before an irrelevant document was returned.

The term recall has been used to indicate the number of documents which were retrieved before all four extremely correlated documents were returned.

The term *accuracy* has been used as an extension of recall. It has only been used where the recall-value and the precision-value were identical. Therefore it includes a measure of quality. Traditionally speaking, it is the precision at 100% recall. This term has been used as principal retrieval measure instead of the traditional terms, because in a real filtering situation the one disastrous event for a user is missing valuable information and not be able to know it. Therefore, the user must be offered an ordered database. This way, personal view thresholds can be set, making it possible to always retrieve all relevant information. Because of this view, precision and recall ratios become less relevant as retrieval measurements. The user expects 100% recall. At what position in the ranking this is achieved, should in the end be for the user to decide. This approach to retrieval measurement is captured in the term accuracy.

Correlation calculation

As measures of correlation, three values have been calculated to determine the optimum Precision and Recall ratio:

- The average error, i.e. the cumulative Euclidean distance to the best matching unit in the neural net for all n-grams in the document, divided by the number of n-grams in the document. This can be thought of as a negative filter, for correlation in this concept is a result of distance calculations. Its value is composed out of information of all n-grams, which results in a global measurement of a document.

- The perfect hit rate, i.e. the cumulative number of n-grams in the document of which its Euclidean distance to the best matching unit is smaller than the hit threshold, divided by the number of n-grams in the document. This can be thought of as a positive filter, for correlation in this concept is a result of counting hits. Its value is not composed out of information of all n-grams, which can result in a global as well as a local measurement of a document. If, for example, only one section in a document correlates to the feature map representation, the document can still be retrieved.

- The average hit error, i.e. the cumulative Euclidean distance to the best matching unit in the neural net for all n-grams in the document of which its Euclidean distance to the best matching unit is smaller than the hit threshold, divided by the number of n-grams in the document. This can be thought of as a positive-negative filter, for it is a fusion of a positive and a negative filter. Correlation in this concept is a result of valuating hits by distance calculations. As in the positive filter, its value is not composed out of information of all n-grams, which can result in a global as well as a local measurement of a document. In the example at b), this positive-negative filter could also clarify how well that section of that document correlates to the feature map.

Expectations

After training, the representation of the full-text query on the feature map was expected to be less accurate than the artificial query's representation, because the full-text query contains a lot of noise, even after passing the input filter. Therefore, the full-text query was expected to be less accurate in the extraction process as well. [artificial +, full-text -]

If the generalisation factor becomes too high, the query's discriminating features will fade too much. Then no accurate correlation between the data flow and the query representation can be distinguished anymore. [generalisation factor +, accuracy -]

With the context size, the information value of a pattern increases exponentially. In other words, the patterns in the feature map as well the patterns of the data flow are more distinctive when the value of the context size is high. [context size +, accuracy +]

In the case of the full-text query, the inclusion of the space as a character to incorporate word adjacencies results in a much higher number of possible patterns in the data stream, while the number of actual patterns in the query does not increase that much, because relatively few combinations appear in the query. This will decrease the perfect hit-probability and thus increase a perfect hit's information value. Therefore, distinguishability increases between relevant and non-relevant documents. [space +, accuracy +]

The average error calculation should serve as a general indication of document relevancy, because of its insensitivity to perfect hits. The perfect hit rate and the average hit error are possible optimisation options, which should at least do well in the case of the full-text query with inclusion of the space as a character.

Execution of the evaluation experiments

The evaluation of the FILTER prototype has been conducted in two stages. First, many different parameter settings were tried in a semi-random search fashion. The goal of this phase was to investigate the effects of the parameters on the behaviour of the system. The results obtained in this phase are described under the heading "Preliminary Results". Second, the more promising regions of the parameter space were isolated and experiments were conducted to get peak performance out of the system. These results are described under the heading "Additional Results".

Preliminary Results

In this section the preliminary results table (Table 10.7 a,b) is globally reviewed, from left to right and from top to bottom. For a good understanding of this table, it should be noted that there are 14 documents which may be retrieved to continue the state of maximum precision: C1, C2, C5, C8, C3, C4, C6, C7, C11, C14, C15, C16, C17, W67. The first 4 documents, however, must be retrieved to reach the state of maximum recall.

TABLE 10.7 A: PRELIMINARY RESULTS: TABLE ABBREVIATIONS.

Settings:	
FTQ	= Full Text Query
AQ	= Artificial Query
Digit 1	= Generalisation factor
Digit 2	= Context size
Digit 3	= Space as character
Digit 4,5,6= Hit threshold (its float value only)	
Q. error:	Average error of the query itself, i.e. the complement of the maximum map activity
D. error:	Average error of the best matching document in the data set
Px:	Maximum Precision document count with hitlist sorted on *x*
Rx:	Maximum Recall document count with hitlist sorted on *x*
Ex:	Error document count at maximum recall with hitlist sorted on *x*
*x*1:	Count with hitlist sorted on Average error
*x*2:	Count with hitlist sorted on Perfect hit rate
*x*3:	Count with hitlist sorted on Average hit error

TABLE 10.7 B: PRELIMINARY RESULTS TABLE.

Settings	Q. error	D. error	P1	R1	E1	P2	R2	E2	P3	R3	E3
FTQ23020	0.119695	0.177454	1	11	2	2	13	6	1	80	70
FTQ43020	0.159510	0.206315	2	14	6	4	18	9	1	37	26
FTQ63020	0.181064	0.217626	1	11	2	1	11	4	1	18	11
FTQ25020	0.179135	0.259371	1	13	3	5	10	2	3	9	3
FTQ45020	0.215698	0.275734	2	14	4	4	13	5	2	15	6
FTQ65020	0.234187	0.287028	1	14	4	4	8	1	1	10	3
FTQ27020	0.219438	0.301106	1	11	3	*6*	*6*	*0*	*5*	*5*	*0*
FTQ47020	0.249470	0.310746	1	15	5	*7*	*7*	*0*	*7*	*7*	*0*
FTQ67020	0.268693	0.318401	1	14	5	*6*	*6*	*0*	*6*	*6*	*0*
FTQ23120	0.120582	0.168126	6	6	0	2	14	5	1	70	59
FTQ43120	0.155662	0.200007	3	8	1	1	15	8	1	34	25
FTQ63120	0.182588	0.217062	2	11	2	1	21	12	1	34	25
FTQ25120	0.183954	0.252867	3	7	1	6	8	1	3	10	4
FTQ45120	0.218768	0.273276	4	8	2	7	7	0	2	6	1
FTQ65120	0.240167	0.286132	3	7	1	4	9	1	2	8	1
FTQ27120	0.228325	0.300309	4	8	2	8	8	0	5	8	1
FTQ47120	0.257845	0.312278	4	8	2	6	6	0	6	6	0
FTQ67120	0.276522	0.321395	4	10	3	6	6	0	6	6	0
AQ23020	0.202122	0.226122	4	15	5	2	28	16	2	58	45
AQ43020	0.232747	0.249897	3	18	9	4	28	16	4	56	43
AQ63020	0.253562	0.267204	2	30	18	2	20	9	1	30	18
AQ25020	0.268683	0.295165	3	12	3	7	7	0	6	8	1
AQ45020	0.284676	0.308447	3	14	5	6	6	0	4	8	2
AQ65020	0.299796	0.320611	2	25	15	8	8	0	5	8	1
AQ27020	0.296442	0.326402	2	26	15	7	7	0	7	7	0
AQ47020	0.315326	0.337489	1	30	19	8	10	1	7	11	2
AQ67020	0.328301	0.344352	2	37	25	6	13	3	5	16	6

Figure 10.8 presents the precision and recall graph for the full-text query with space as a character. Only the accurate results are visualised here. The actual accuracy value is contained within these charts at the point the fourth and last highly relevant document, i.e. n=4, has been retrieved.

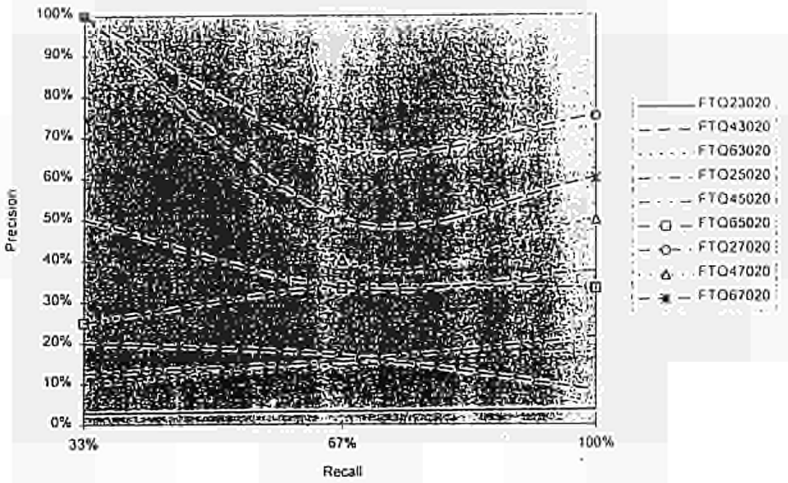


FIGURE 10.8: PRECISION VERSUS RECALL (FULL-TEXT QUERY, SPACE AS A CHARACTER, SORTED ON AVERAGE HIT ERROR) - PRELIMINARY RESULTS:

The difference between the average error of the query and the average error of the best matching document is significantly smaller in the case of the artificial query than in the case of the full-text query. Whenever extremely accurate hit threshold sensitive results were returned, the average error of the query was always higher than the hit threshold.

When sorted on the average error, the full-text query without space as a character did not do so well. Even its best results were not acceptable. The full-text query with space as a character did do better. The results even turned out to be very accurate, when the generalisation factor as well as the context size were set low. The artificial query did not do well at all.

When sorted on the perfect hit rate, the full-text query without space as a character did do very well when, but only when, the context size was set high. In these settings a state of extreme accuracy was reached. The generalisation factor had become insignificant at this point. The full-text query, with space as a character, did even do slightly better. In these settings the state of extreme accuracy was already more or less reached using only a medium context size, but at this point the generalisation still played a significant role. The artificial query also became extremely accurate when the context size was set medium. With a high context size, the accuracy decreased again. At this point the generalisation factor became relevant again.

When sorted on the average hit error, the results were also extremely good in some configurations, but in general less stable than the results for the perfect hit rate. These two hit

threshold-sensitive correlation calculations seem to react in the same way to parameter variations, but the perfect hit rate yields somewhat better results.

Preliminary Analysis

The artificial query is noiseless. Because of this property, any generalisation of its data will therefore reduce its discriminating features and thus decrease the accuracy of the query representation. This is reflected in the small difference between the average error of the query and the average error of the best matching document. This effect can be suppressed though, by using a higher context size. The information value of a query pattern increases more than it is decreased by generalisation. However, when the context size is set too high, too many keywords with a length, smaller than the context size, are incorrectly represented by the internal keyword concatenation.

The generalisation factor must, in general, not be set too high to avoid significant reduction of its discriminating features. However, this parameter is only of importance when the context size is not set high.

Without the inclusion of space as a character, the information value of the patterns in the full-text query can become too low. Also, by inclusion of the space as a character and thus the inclusion of adjacent word relations, the relative number of relevant patterns increases significantly[40]. Therefore, with the space as a character, the information value increases as well as the accuracy of the query's representation in the feature map.

The context size must not be set too high to avoid inaccurate representation of discriminating features[41]. However, if a good representation of only a few, but relevant, patterns has been formed in the feature map and the context size has been set high, the generalisation factor and the space as a character become relatively insignificant. Although the average error will be of a relatively indeterminate nature, the perfect hits these few patterns will cause, will be extremely informative *when the hit threshold is set lower than the maximum map activity and the hitlist is sorted on the perfect hit rate.*

---

[40] For example, the string _DESKTOP_COMPUTER_ is decomposed into 16 fully correct trigrams, whereas DESKTOPCOMPUTER is decomposed into only 11 fully correct trigrams.

[41] For the query used in this evaluation, discriminating features are for example DELL, CHIP, INTEL, etc.

The instability of the average hit error-sorted hitlists can be explained by the paradoxical nature of these values. The results are only accurate when the hit threshold is set relatively low. This means there are relatively few perfect hits. Thus the intentional effects are suppressed. In other words, this correlation calculation only performs well when it is transformed into a perfect hit rate imitation. Therefore, the average hit error-sorted results are considered not to be relevant.

To validate this analysis, new settings have been evaluated. There has been focused on configurations with a low context size to minimise execution times.

Additional Results

In this section the additional results table is globally reviewed, from top to bottom. The additional results can be found below in Table 10.8 (see Table 10.7 for the abbreviations used).

TABLE 10.8: ADDITIONAL RESULTS TABLE

Settings	Q. error	D. error	P1	R1	E1	P2	R2	E2	P3	R3	E3
FTQ27030	0.219438	0.301106	1	11	3	1	16	9	1	20	12
FTQ27010	*0.219438*	*0.301106*	1	11	3	*6*	*6*	*0*	*6*	*6*	*0*
FTQ23012	0.122862	0.180017	1	11	2	2	7	1	1	10	4
FTQ23009	0.122862	0.180017	1	11	2	2	9	1	1	10	3
FTQ23006	*0.122862*	*0.180017*	1	11	2	*6*	*6*	*0*	*7*	*7*	*0*
FTQ23109	*0.120582*	*0.168126*	*6*	*6*	*0*	*6*	*6*	*0*	4	7	1
FTQ23106	*0.120582*	*0.168126*	*6*	*6*	*0*	*4*	*4*	*0*	*4*	*4*	*0*
FTQ43107	0.155662	0.200007	3	8	1	1	23	11	1	36	24
FTQ43111	*0.155662*	*0.200007*	3	8	1	*6*	*6*	*0*	5	9	2
AQ13020	0.176395	0.205944	5	10	2	4	19	9	1	47	35
AQ130176	0.176395	0.205944	5	10	2	5	15	5	1	39	29
AQ130132	0.176395	0.205944	5	10	2	9	10	1	2	28	18
AQ130088	*0.176395*	*0.205944*	5	10	2	*8*	*8*	*0*	7	15	5
AQ130044	*0.176395*	*0.205944*	5	10	2	*8*	*8*	*0*	*7*	*7*	*0*
AQ130022	*0.176395*	*0.205944*	5	10	2	*8*	*8*	*0*	*8*	*8*	*0*
AQ15020	0.237150	0.279872	3	13	3	7	11	2	3	13	4
AQ230155	0.202122	0.226122	4	15	5	2	18	7	1	31	19
AQ230101	*0.202122*	*0.226122*	4	15	5	*9*	*9*	*0*	*9*	*9*	*0*
AQ230005	0.202122	0.226122	4	15	5	4	13	6	4	12	5

Figure 10.9 presents the precision and recall graph for the full-text query with space as a character. Only the accurate results are visualised here. The actual accuracy value is contained within these charts at the point the fourth and last highly relevant document, i.e. n=4, has been retrieved. Since only the hit threshold parameter has been varied and the same settings were used as in the preliminary results, the average error accuracy does not appear here.

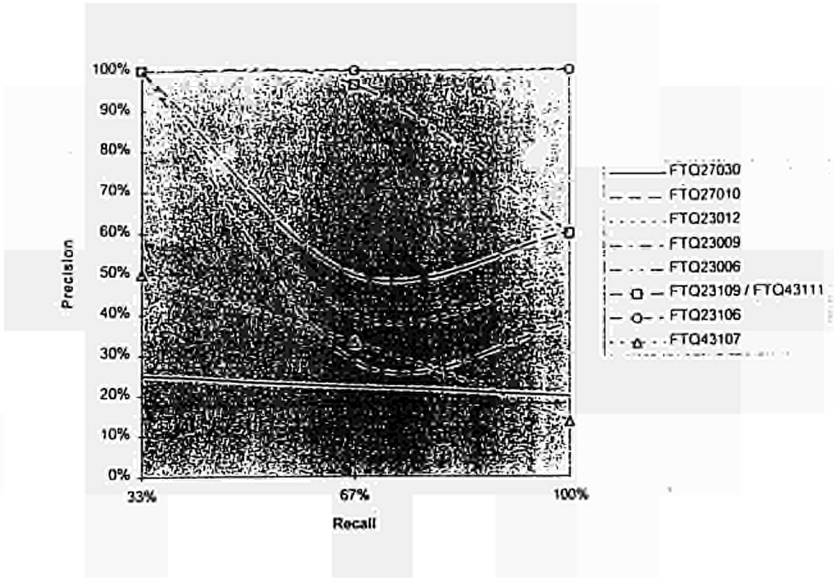


FIGURE 10.9: PRECISION VERSUS RECALL (FULL-TEXT QUERY, SPACE AS A CHARACTER, SORTED ON PERFECT HIT RATE) - ADDITIONAL RESULTS:

To validate the assumption that the hit threshold should be set lower than the maximum map activity, one extremely accurate configuration[42] with a hit threshold lower than the maximum map activity, was re-evaluated with a value, higher than the maximum map activity. This resulted in highly inaccurate outcomes.

Then, the effective range had to be investigated. Therefore the same setting was re-evaluated again, now with a hit threshold, set at more than 50 percent below the minimum map error. The results were extremely accurate.

Next, this experiment was repeated in more detail with one of the highly inaccurate results where a low context size had been used[43]. Accuracy increased as the hit threshold was set lower. When a hit threshold was used of 50 percent below the minimum map error, the state of extreme accuracy was reached.

At this point the experiment was transferred to the only accurate average error-sorted result[44]. Not only did the hit threshold-modifications react identically, but using a hit threshold of 50 percent below the minimum map error, a **perfect** hitlist was retrieved here. Not only were the

---

[42] Settings properties: full-text query, low generalization factor, high context size, no space as character.

[43] Settings properties: full-text query, low generalization factor, low context size, no space as character.

[44] Settings properties: full-text query, low generalization factor, low context size, space as character.

four most important documents retrieved first, but the next six documents were also significantly correlated.

Until then, the generalisation factor had been kept constant at a low value. Knowing more about the hit threshold, the influence of the generalisation factor was examined in relation to the hit threshold[45]. When hit threshold was used of 50 percent below the maximum map activity, the results were not good at all. However, when a hit threshold of 25 percent was used, the results became extremely accurate.

This left only the artificial query to be optimised, because accurate results with the artificial query would make a comparison with index-based information retrieval systems easier. First an experiment with a configuration with no generalisation factor[46] was carried out in detail. Again, when a hit threshold of 50 percent below the maximum map activity was used, the results became accurate. The hit threshold was then set even lower, until a hit threshold of 88 percent below the maximum map activity. The state of accuracy was continued.

Finally, this experiment was transferred to one of the original artificial settings[47]. Again, when a hit threshold of 50 percent below the maximum map activity was used, the results became accurate. However, when a value of 75 percent below the maximum map activity was used, the results became highly inaccurate again.

Additional Analysis

The hit threshold is the most essential parameter. In general, a value of 50 percent below the maximum map activity gives accurate results. However, as the generalisation factor increases, which causes a decrease in the accuracy of the map's query representation, the hit threshold-value should also increase.

With the context size, robustness and execution time increase. But, *all* preliminary configurations should be optimisable by merely adjusting the hit threshold-value.

The addition of the space as a character increases the information value of the patterns and thus the retrieval quality.

---

[45] Settings properties: full-text query, medium generalization factor, low context size, space as character.

[46] Settings properties: artificial query, no generalization factor, low context size.

[47] Settings properties: artificial query, low generalization factor, low context size.

The artificial query performs best when it is not compressed.

Comparison

Although the data set was also evaluated with ZyIMAGE, a contemporary index-based information retrieval system, it is not straightforward how a fair comparison can be made with FILTER.

First of all, the data set was composed with ZyIMAGE. This makes a comparison by definition unfair. Also, the number of documents in the data set is too small to compare the systems adequately. A second problem for a fair comparison is the semantic nature of the query's subject. All words starting with the string COMPUT seem to belong to the semantic class of computer terms. Index-based information retrieval systems perform best when these kind of subjects are to be retrieved, because they search for more or less exact matches. The neural filter's performance is subject-insensitive, because it does not search for exact matches, but calculates a correlation.

Also, the best neural filter results were obtained with the full-text query. If the artificial query had yielded the best results, a more valuable comparison could have been made by using exactly the same query in both systems.

Having emphasised the relative importance of any comparison made between these systems, the results, obtained with ZyIMAGE, are reviewed and a comparison is made with the results, obtained with FILTER.

A few Boolean queries were evaluated in the same way as the neural filter queries were evaluated. The approach, taken to retrieve the most accurate results, was to extend the original elementary query by including a few more keywords with the operators AND & OR in such a way that this Boolean query would still represent the semantical core of the full-text query. Perfect retrieval was achieved when the original query was extended with one or two keywords of the set {LINE, DESKTOP, DELL}.

Next, the artificial query itself was evaluated as a quorum query. First without the keyword repetitions, making it an unweighted quorum query with a varying quorum value. The results were highly accurate, until the query became too informative. Next, some the most frequent keyword repetitions were translated into a weighted quorum query. Also Boolean-quorum mixtures were evaluated. In all cases an increase of the information value of the query resulted in a decrease of accuracy.

In all cases, the fuzzy searches returned none or too few documents with these Boolean queries.

In this comparison, with this data set and this query, both in the neural filter and the traditional information retrieval system using Boolean queries, perfect retrieval could be achieved. The artificial query seemed to contain too much information for the index-based system. It only worked well when the artificial query was more or less reduced to a big disjunction of keywords. This high information density was also a problem in the neural filtering environment, but here this problem could be solved quite easily by fine-tuning one or two parameters. In other words, although results were identical in this comparison, the neural filter seems more flexible and more robust.

The results of the comparison experiments are listed in detail in Table 10.9 a,b,c below.

TABLE 10.9 A: COMPARISON WITH ZYIMAGE - ABBREVIATIONS

- F: Fuzzy degree used
- P: Maximum Precision document count
- R: Maximum Recall document count
- E: Error document count at maximum recall

TABLE 10.9 B: ZYIMAGE BOOLEAN QUERIES RESULTS TABLE

Query	F	P	R	E	Comment
COMPUT*	0-4	8	8	0	data composition query
COMPUT* OR DELL	0	6	6	0	
	1-4	8	8	0	
COMPUT* OR DELL*	0-4	6	6	0	
(COMPUT* OR DELL) AND (NOT KEYB*)	0-4	6	-	-	1 essential doc was not found
(COMPUT* OR DELL) AND (NOT SOFTW*)	0	5	5	0	
	1-4	6	6	0	
(COMPUT* OR DELL*) AND (LINE OR DESKTOP)	0	4	4	0	
	1-4	-	-	-	search generated no hits
(COMPUT* OR DELL) AND (LINE OR DESKTOP)		4	4	0	query was not the best hit
	1-4	-	-	-	search generated no hits
COMPUT* AND (LINE OR DESKTOP OR DELL*)	0	4	4	0	
	1-4	2	-	-	only 2 docs were retrieved
COMPUT* AND (LINE OR DESKTOP OR DELL)	0	4	4	0	query was not the best hit
	1-4	-	-	-	search generated no hits

TABLE 10.9 C: ZYIMAGE QUORUM QUERIES RESULTS TABLE

Query	F	P	R	E
1 of {Dell, Desktop, Computers, OptiPlex, line, introduction, high, performance, Pentium, microprocessors, computer, price, barrier, configured, system, speed, megahertz, Intel, chip, Doug, MacGregor, chips, machines, 486, models, power, management, networking, plug, play}	0	5	5	0
1 of {...}	1-4	1	-	-
2 of {...}	0	5	5	0
2 of {...}	1-4	-	-	-
3 of {...}	0	5	5	0

4 of {...}	0	5	5	0
5 of {...}	0	6	6	0
6 of {...}	0	1	-	-
2 of {Dell*, Computer*, Desktop} AND 1 of {OptiPlex, line, introduction, high, performance, Pentium, microprocessors, computer, price, barrier,configured, system, speed, megahertz, Intel, chip, Doug, MacGregor, chips, machines, 486, models, power, management, networking, plug, play}	0	4	-	-
1 of {Dell*, Computer*, Desktop} AND 1 of {OptiPlex, line, introduction, high, performance, Pentium, microprocessors, computer, price, barrier,configured, system, speed, megahertz, Intel, chip, Doug, MacGregor, chips, machines, 486, models, power, management, networking, plug, play}	0	5	5	0
Dell* AND Computer* AND Desktop AND 1 of {OptiPlex, line, introduction, high, performance, Pentium, microprocessors, computer, price, barrier,configured, system, speed, megahertz, Intel, chip, Doug, MacGregor, chips, machines, 486, models, power, management, networking, plug, play}	0	2	-	-
Dell* AND Computer* AND 1 of {Desktop, OptiPlex, line, introduction, high, performance, Pentium, microprocessors, computer, price, barrier,configured, system, speed, megahertz, Intel, chip, Doug, MacGregor, chips, machines, 486, models, power, management, networking, plug, play}	0	2	-	-
Computer* AND 2 of { Dell, Desktop, OptiPlex, line, introduction, high, performance, Pentium, microprocessors, price, barrier,configured, system, speed, megahertz, Intel, chip, Doug, MacGregor, chips, machines, 486, models, power, management, networking, plug, play}	0	6	6	0

10.4 Discussion

Two important consequences can be drawn from the evaluation:

- The neural filter yields highly accurate results when the parameters are set properly. In the prototype, parameters can be calculated automatically by the hints mechanism. Since this process does not require any additional fine-tuning, it only takes minimum preparation time.

- By using FILTER's table map, maximum execution speed possible can be maintained without compromising retrieval accuracy, which is essential in current awareness applications[48].

Although the neural filter is likely to exhibit more flexible and more robust behaviour than a index-based information retrieval system with respect to a query, as has been pointed out in the comparison section, this primarily holds in a static query environment. In such an environment, a query functions as a user profile. All incoming data passes a series of profiles, all text parts are extracted accordingly and the results are stored in databases.

However, a problem arises when new queries are added. In that case, the system will have to process the whole corpus to obtain an initial update, which can take quite some time, because the incoming data is not made quickly accessible by some sort of indexing. This is the major drawback of the neural filter-algorithm. All data of a text part is needed to determine its correlation with respect to a query, instead of merely locating occurrences of query-strings in the generated index, as is, roughly speaking, the case in index-based retrieval systems.

The optimum corpus preparation for the neural filter system would probably be to store a compressed vector representation of each text part which still contains all data patterns to maintain maximum performance and minimise data storage overhead. By adding the number of occurrences in the text part to each data pattern, all pattern repetitions can be eliminated. This way, the character-to-vector translations and all iterative cycles are eliminated from the corpus extraction process. Although these measures seriously affect the flexibility of the

---

[48] The data set, used in the evaluation, was processed in about 8 minutes on a 66 Megahertz-486DX2 PC. This means that the FILTER Prototype processes 6 Mb per hour, while maintaining accurate retrieval.

system, because it implies fixed pattern coding parameters, this does not necessarily have to be a problem, because generally applicable parameter-settings have been established in the evaluation.

The neural filter could also be added to existing current awareness applications in a data fusion environment to improve retrieval quality. The idea behind data fusion is that any combination of methods will yield better results than any method applied stand-alone, because each method examines its input from a different perspective, which results in a different output.

In the neural filter, retrieval consists of calculating the correlation of all data patterns in relation to a rigid query representation. In an index-based filter, retrieval consists of locating the query-components in a rigid data index. In other words, the task is approached from quite opposite perspectives. By combining the results of such a compound analysis, more accurate and more robust results are likely to be obtained.

This report has shown that the neural filter can contribute significantly to the class of real-time filtering applications as a high quality full-text search method**, especially in a data fusion environment.**

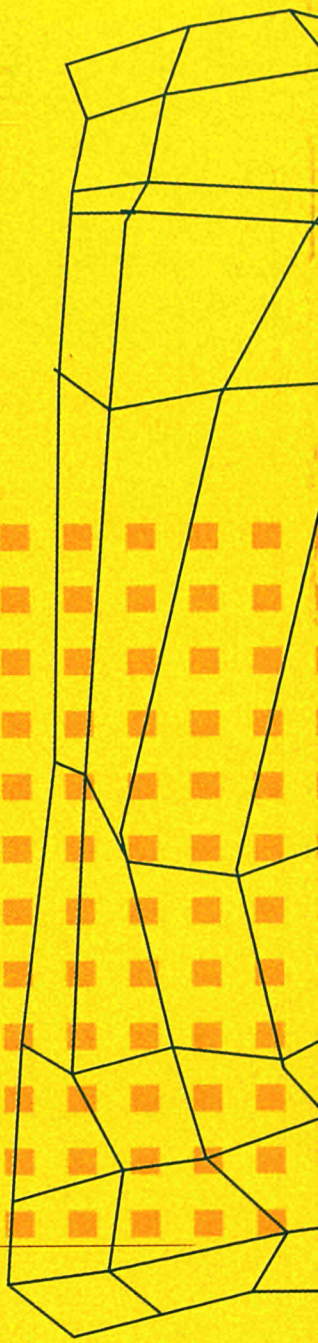
15

CD-NA-16264-EN-C

Results of a study done
by MSC Information Retrieval Technologies BV,
the Netherlands, for the European Commission,
Telematics for libraries

NOTICE TO THE READER

All scientific and technical reports
published by the European Commission
are announced in the monthly periodical
'euro abstracts'.
For subscription (1 year: ECU 63)
please write to the address below.



Price (including VAT) in Luxembourg: ECU 33

ISBN 92-827-4690-9

OFFICE FOR OFFICIAL PUBLICATIONS
OF THE EUROPEAN COMMUNITIES

L-2985 Luxembourg