FILTER Prototype Report

a neural filtering environment
Table of Contents

TABLE OF CONTENTS........................................................................................................... i
PREFACE............................................................................................................................... iv

INTRODUCTION...................................................................................................................... v

BACKGROUND....................................................................................................................... 6
1.0 INTRODUCTION .................................................................................................................. 6
1.1 INFORMATION RETRIEVAL ............................................................................................... 6
1.2 ARTIFICIAL NEURAL NETWORKS ...................................................................................... 8
  1.2.1 Kohonen feature map .................................................................................................. 8
  1.2.2 Alternatives ............................................................................................................... 9
1.3 NEURAL FILTER ............................................................................................................... 10
1.4 SUMMARY ....................................................................................................................... 11

PROTOTYPE............................................................................................................................. 12
2.0 INTRODUCTION .................................................................................................................. 12
2.1 PROPERTIES ..................................................................................................................... 12
  2.1.1 Flexibility ................................................................................................................... 12
  2.1.2 Performance .............................................................................................................. 13
    2.1.2.1 Dynamic k-d tree ................................................................................................. 13
    2.1.2.2 Table map .......................................................................................................... 14
  2.1.3 Visualisation ............................................................................................................... 15
    2.1.3.1 Textual visualisation ........................................................................................... 15
    2.1.3.2 Graphical visualisation ....................................................................................... 15
  2.1.4 Accessibility ............................................................................................................... 16
2.2 OVERVIEW ...................................................................................................................... 17
2.3 SUMMARY ....................................................................................................................... 19

EVALUATION............................................................................................................................ 20
3.0 INTRODUCTION .................................................................................................................. 20
3.1 PREPARATION .................................................................................................................... 20
  3.1.1 Data set ...................................................................................................................... 20
  3.1.2 Queries ...................................................................................................................... 21
    3.1.2.1 Full-text query .................................................................................................... 21
    3.1.2.2 Artificial query .................................................................................................. 21
  3.1.3 Settings ...................................................................................................................... 22
  3.1.4 Definitions .................................................................................................................. 23
  3.1.5 Correlation calculation .............................................................................................. 23
3.2 EXPECTATIONS .................................................................................................................. 24
3.3 EXECUTION ....................................................................................................................... 25
  3.3.1 Preliminary results ..................................................................................................... 25
  3.3.2 Preliminary analysis .................................................................................................. 26
  3.3.3 Additional results ..................................................................................................... 27
  3.3.4 Additional analysis ................................................................................................... 28
3.4 COMPARISON .................................................................................................................. 29
3.5 SUMMARY ....................................................................................................................... 30

DISCUSSION............................................................................................................................. 32
4.0 INTRODUCTION .................................................................................................................. 32
4.1 CONCLUSIONS .................................................................................................................. 32
4.2 FUTURE RESEARCH .................................................................................. 32
4.2.1 Corpus representation ............................................................................. 32
4.2.2 Data fusion ............................................................................................... 33
4.3 SUMMARY .................................................................................................... 33

REFERENCES ................................................................................................. 34

APPENDICES .................................................................................................... 35

BACKGROUND [APPENDICES] ......................................................................... 36
1.3 NEURAL FILTER .......................................................................................... 36
   Principle ........................................................................................................... 36
   n-gram analysis method .................................................................................... 36
   Algorithm (schematic version) .......................................................................... 37

PROTOTYPE [APPENDICES] ............................................................................ 38
2.0 INTRODUCTION .......................................................................................... 38
2.1.1 Flexibility .................................................................................................. 38
2.1.2.1 Dynamic k-d tree .................................................................................. 40
   Visualisation ................................................................................................... 40
2.1.3.1 Textual visualisation ............................................................................. 40
   Feature map ..................................................................................................... 40
   Vector map ....................................................................................................... 41
   Object frequency analysis .............................................................................. 42
2.1.3.2 Graphical visualisation ......................................................................... 43
   State of self-organisation ................................................................................ 43
   Interneuronal distances .................................................................................. 44
   Object distribution (best objects possible) ...................................................... 45
   Object distribution (best objects in query) ..................................................... 46
   Error recording ............................................................................................... 47
   Activity recording ........................................................................................... 48
2.2 OVERVIEW .................................................................................................. 49
   FILTER workspace .......................................................................................... 49
   FILTER toolbar, after loading data, off line .................................................... 49
   FILTER toolbar, during the extraction process ............................................. 50
   Command Settings-Parameters ....................................................................... 50
   Command Settings-Preferences ....................................................................... 52
   Command Settings-Files and Paths ................................................................. 53
   Command Settings-Parameters-Hints ............................................................. 54
   Command Settings-Parameters-Demo ............................................................ 55
   Command Process-Table Map Menu ............................................................... 55
   Command Hitlist-Sort Hitlist .......................................................................... 56
   Command Settings-Evaluation Mode ............................................................... 56
   FILTER menus ............................................................................................... 57
   FILTER statusbar ........................................................................................... 59

EVALUATION [APPENDICES] ........................................................................... 61
3.1.1 Data set composition ................................................................................ 61
   Query 1: WAR* ............................................................................................... 61
   Query 2: EC ..................................................................................................... 63
   Query 3: COMPUT* ......................................................................................... 63
   Documents with multiple occurrences ......................................................... 63
   Correlation assignment ................................................................................... 63
3.1.2.1 Full-text query ...................................................................................... 64
   C1 (1F7.TXT) .................................................................................................. 64
3.1.2.2 Artificial query ...................................................................................... 65
   C1', derived from C1 (1F7.TXT) ...................................................................... 65
3.1.3 Settings composition ................................................................. 65
  Used configurations in the preliminary evaluation process .................. 65
  Example of a settings configuration (FTQ23106.SET) ......................... 66
3.2 EXPECTATIONS ............................................................................ 67
  Input filter: common words ............................................................ 67
  Input filter: common endings ......................................................... 68
  Full-text query’s internal data flow ................................................ 68
  Artificial query’s internal data flow ................................................ 69
3.3.1 Preliminary Results .................................................................. 70
  Measurement .................................................................................. 70
  Table abbreviations ....................................................................... 70
  Table of results .............................................................................. 70
  Precision and recall, sorted on average error^i .................................. 71
  Precision and recall, sorted on perfect hit rate^i ............................... 72
  Precision and recall, sorted on average hit error^ii ............................ 72
  Precision and recall, sorted on average error^ii ............................... 73
  Precision and recall, sorted on perfect hit rate^iii ............................. 74
  Precision and recall, sorted on average hit error^iii .......................... 74
  Precision and recall, sorted on perfect hit rate^iii ............................. 75
  Precision and recall, sorted on average hit error^iii .......................... 75
3.3.3 Additional Results ..................................................................... 76
  Measurement .................................................................................. 76
  Table abbreviations ....................................................................... 76
  Table of results .............................................................................. 76
  Precision and recall, sorted on perfect hit rate^i ................................ 77
  Precision and recall, sorted on average error^ii ............................... 78
  Precision and recall, sorted on perfect hit rate^ii ............................. 78
  Precision and recall, sorted on average hit error^iii .......................... 79
3.4 COMPARISON .............................................................................. 79
  Measurement .................................................................................. 79
  Table abbreviations ....................................................................... 79
  Table of results (ZyIMAGE Boolean queries) ..................................... 80
  Table of results (ZyIMAGE quorum queries) .................................... 80

INDEX .......................................................................................... 83
Preface

FILTER\(^1\) is a prototype for the EC Libraries Program *Neural Networks and Information Retrieval in a Libraries Context*. The prototyping has been carried out for MSC Information Retrieval Technologies BV. Its background, development and evaluation is described in this report.

This project has been supervised by Prof. Dr. Ir. R.J.H. Scha and Dr. Ir. J.C. Scholtes, whom I would hereby like to thank for all their time invested in me. I would also like to thank Drs. E. Brinkhuis for his remarks on the preliminary version of this report.

Marco René Spruit,
Department of Computational Linguistics,
Faculty of Arts,
University of Amsterdam,


\(^{1}\) The picture on the front cover of this report is the application icon of the FILTER Prototype.
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\(^1\) or the filter principle.

The amount of information that has to be 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\(^2\). 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 report presented here has been divided into two sections. The main section describes the work that has been carried out in a highly compressed manner. 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. Then, it is explained how the prototype has been evaluated and what conclusions can be drawn based on this evaluation. The main section is concluded with a brief recapitulation of the main consequences and some directions for future research. The extensive appendices section contains additional clarification and background information to provide a more profound insight in the project. This section uses parts of the hierarchical structure of the main section to provide implicit cross-referencing. This entire report is also available within the prototype itself as part of the on-line documentation.

\(^1\) Selective Dissemination of Information (SDI). For readability, abbreviations are only used in footnotes and appendices.

\(^2\) Examples of analogue information sources are paper and microfilm. Examples of digital information sources are CD-ROM’s, on-line services and computer networks.
Background

1.0 Introduction

This chapter provides a basic theoretical background necessary to obtain some insight in the context of this project. It has been divided into three main sections.

The first section introduces the idea of information retrieval\(^1\) from an index-based perspective\(^2\).

The second section explains the concept of artificial neural networks\(^3\). 1.2.1 describes the Kohonen feature map\(^4\) and its main properties. 1.2.2 discusses a few alternative neural architectures that also could be used in information retrieval.

The third section approaches the filter principle from a neural information retrieval perspective. This chapter is concluded with a summary in 1.4.

1.1 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 that is investigated with dynamic queries. This is known as free-text search or document retrieval.
- A dynamic database environment that 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 that 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 out of keywords with optional wildcards. Its internal relations

---

1 Information Retrieval (IR).
2 The phrase index-based perspective refers to all information retrieval methods based on a quickly accessible internal representation of the database. In other words, methods that do not derive an internal representation of the query.
3 Artificial Neural Network (ANN).
4 The Kohonen Feature Map (KFM), developed by T. Kohonen, is also referred to as Topological Feature Map (TFM).
can be controlled by logical and statistical operators\(^5\). 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 that 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 to be quite difficult to formulate a query that accurately corresponds to his or her intentions.
- Only documents that contain the query-keywords can be retrieved\(^6\). It has no significant ability to generalise over a query and cannot make incomplete matches intelligibly\(^7\).
- No real context can be incorporated\(^8\).

The field of information retrieval would be greatly indebted to a method that could incorporate these shortcomings without slowing down.

---

\(^5\) 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 \(\{k_1, k_2, \ldots\}\). This means that \(n\) keywords of the keyword-set must be in the document. See also Appendix 3.4.

\(^6\) Thesauri, or synonym-vocabularies, can be included in contemporary IR applications to optimise retrieval. However, this is not a real solution. The danger of retrieval-overkill increases significantly.

\(^7\) 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.

\(^8\) There do exist some context sensitivity-imitation techniques. An example is the binary proximity operator \(A\) within \(n\) words of \(B\). This operator 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.
1.2 Artificial neural networks

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

- Distributed data representation, i.e. $x$ objects are represented by $y$ neurons. 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 neurons increases.

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

1.2.1 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 neuron, i.e. a set of input fibres or sensors or a vector representing a co-ordinate in the vector space. The data that 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 that 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 neurons that all have the same number of weights. The activation of a neuron resulting from an input activation is interpreted as a measure of correlation. Therefore, the neuron best representing the input activation can be determined by finding the neuron with the highest activity. In other words, the neuron best representing the input vector can be determined by finding the map vector with the minimum

---

9 An object can, for example, be a natural language character or word.

10 Architectures that do not have this property are not considered neural, but statistical table models.
mathematical distance\(^{11}\) with respect to the input vector. This neuron is called the best matching unit\(^{12}\).

Once this neuron has been found, neurons 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. A topological map emerges that holds related data elements in neighbouring regions because neighbouring neurons are updated with respect to an input vector's best matching unit each training cycle\(^{13}\).

To summarise, the feature map has some additional interesting properties besides the general artificial neural network properties mentioned in 1.2:

- **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. 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.

### 1.2.2 Alternatives

The adaptive resonance theory\(^{14}\) also encapsulates self-organisation and unsupervised training in a more neurobiologically inspired 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.

---

\(^{11}\) The most commonly used mathematical distance in vector space models is the Euclidean distance. It can be calculated by \(\sqrt{x^2 + y^2}\), where \(x\) is the distance between two X co-ordinates and \(y\) is the distance between two Y co-ordinates. This distance has also been used in this prototype.

\(^{12}\) Best Matching Unit (BMU).

\(^{13}\) See Appendix 1.3 for a schematic overview of the training algorithm as part of the filter principle algorithm.

\(^{14}\) The Adaptive Resonance Theory (ART) has been developed by S. Grossberg.
The simple recurrent network\textsuperscript{15} uses a recurrent network where the hidden layer units are fed back into the input layer. Training such a network is also unsupervised. The model implements a high order Markov chain\textsuperscript{16} by using recurrent input fibres. 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 because it is not known what is to be trained in advance. Therefore, the Kohonen feature map has been implemented.

1.3 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 that derives an internal representation of the text. This representation is then matched against a continuous stream of incoming, unstructured data. Optionally, multiple queries can be matched simultaneously\textsuperscript{17}.

Several algorithmic variants are possible depending on the choice of objects that 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\textsuperscript{18}. This variant is known as the $n$-gram analysis method. A $n$-gram is a $n$-length sequence of characters\textsuperscript{19}. The $n$-gram analysis method can be interpreted as a window of size $n$ that 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.

\textsuperscript{15}The Simple Recurrent Network (SRN) has been developed by J.L. Elman.

\textsuperscript{16}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 that are used to predict the next character.

\textsuperscript{17}See Appendix 1.3 for a visualisation of the neural filter principle and a schematic version of the algorithm.

\textsuperscript{18}A dictionary would have been necessary if words would have been used as objects. Such a dictionary could have been generated in advance though by a statistical frequency algorithm. However, in the FILTER prototype language-dependent noise words and noise word endings can be eliminated to optimise performance. But these noise lists could also be generated in advance by a statistical frequency algorithm.

\textsuperscript{19}For example, the set of possible trigrams, i.e. $n=3$, with the word TRIGRAM is \{??T, ?TR, TRI, RIG, GRA, RAM, AM?, M??\}, where ?’s indicate variable context characters.
After training, the input values of texts, mediated through the shifting window, that 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, as mentioned in 1.1.

1.4 Summary

Two types of information retrieval applications were distinguished. Free-text search applications operate in a relatively static database environment that is investigated with dynamic queries. The filtering problem investigated in this project operates in a dynamic database environment that needs to be filtered with respect to relatively static queries. Some serious drawbacks of the index-based approach are the difficulty of formulating one’s intentions, the inability to properly generalise over a query and the lack of intelligible context incorporation.

Artificial neural networks were defined and interesting properties as distributed data representation, robust behaviour and language independency were determined. These properties were considered important in an information retrieval context. The Kohonen feature map has some additional interesting properties. Self-organisation on frequency and context, unsupervised training, topology preservation, relative computational efficiency and implementation ease make the Kohonen feature map the finest neural architecture for the filtering problem.

The neural filter algorithm was reviewed. It was explained why the $n$-gram analysis method was used and how the feature map can be incorporated as a filtering device. This approach was expected to resolve some of the drawbacks of index-based retrieval.
Prototype

2.0 Introduction

This chapter describes the prototype. It has been divided into two main sections.

The first section describes the application properties on which was focused during the prototype development. 2.1.1 explains the importance of flexibility. 2.1.2 describes what is meant with performance. 2.1.3 emphasises the need for visualisations of the feature map. 2.1.4 focuses on the accessibility issue.

The second section gives an overview of the prototype in the form of an imaginary session. This chapter is concluded with a summary in 2.3.

2.1 Properties

Four application properties were considered essential during the prototype development to create a well 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.

2.1.1 Flexibility

Flexibility, as defined in 2.1, stresses the importance of interactive research 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 concept\(^1\). The prototype has been implemented

---

\(^1\) The declaration of the feature map object is given in Appendix 2.1.1 to illustrate the concept of Object-Oriented Programming (OOP).
in C++\textsuperscript{2} because this original Microsoft Windows programming language can provide maximum execution speed possible within such an environment as well as portable code\textsuperscript{3}.

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

### 2.1.2 Performance

The term _accuracy_ in the definition of performance in 2.1 will be used continuously in the evaluation section and will be defined in 3.1.4. For now the general intuitive notion should suffice.

Performance, as defined in 2.1, stresses the importance of execution speed in this type of applications. The importance of the accuracy of the retrieval is here merely implicitly accentuated 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 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\textsuperscript{4}:

- **Read** the incoming data in chunks.
- **Shift** a window over each chunk to retrieve the object patterns.
- **Convert** each object pattern to its vector representation.
- **Search** the nearest neighbour in the feature map for each input vector.

The most time-consuming event in a uniprocessor\textsuperscript{5} 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.

#### 2.1.2.1 Dynamic k-d tree

A dynamic k-d tree was implemented\textsuperscript{6} first. 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

---

\textsuperscript{2} The FILTER Prototype has been implemented with the Microsoft Visual C++ Development System for Windows, Professional Edition, Version 1.5. The implementation has been based on the Microsoft Foundation Classes (MFC) Library.

\textsuperscript{3} If the prototype is to be transferred to another operating system only the user interface might have to be rewritten. Actually, the prototype should run as is within the Windows NT environment as a 32-bit application after recompilation of the code.

\textsuperscript{4} See Appendix 1.3 for the schematic version of the algorithm.

\textsuperscript{5} The conceptually most obvious minimisation of execution time, the implementation in a multiprocessor environment, has been completely ignored in this report for practical reasons.
training, it should also be possible to convert the feature map into this structure after training by deviding the feature map recursively into two equal neuron collections along the axis of greatest range. The data points are stored in the leaf nodes. The search time becomes logarithmically\(^7\) if the tree is in balance when this search technique is used.

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. This led relatively often 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 was worked out.

2.1.2.2 Table map

Hashing is a well known statistical addressing technique. It calculates a function of the input to retrieve the address of the output. In this case it 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 that 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 becomes a repetition of the following four events:

- Read the incoming data in chunks.
- Shift a window over each chunk to retrieve the object patterns.
- Convert each object pattern to its hashing address.
- Get the distance contained in that address for each object pattern.

However, there can be very many possible patterns depending on the context size\(^8\). A two-dimensional hashtable was implemented to enable storage capacity for up to a hundred million distances. 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

\(^6\) See Appendix 2.1.2.1 for a visualisation of a feature map in the form of a tree structure.

\(^7\) The full-search algorithm has a time complexity \(O(N)\), where \(N\) is the number of neurons in the feature map. The tree-search algorithm has a time complexity \(O(\log N)\).

\(^8\) The number of possible patterns is \(O^c\), where \(O\) is the number of objects in the language and \(c\) is the context size.
It takes a lot of memory and preparation time. Therefore, the context size should be kept low if the table map is to be used. In the evaluation section it will be investigated whether a low context size is possible or not in relation to the retrieval accuracy.

2.1.3 Visualisation

Visualisation, as defined in 2.1, stresses the importance of viewing the data from different perspectives to help clarifying 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.
- Graphical visualisation, i.e. a print of relations within an object in an unrestricted format. This is very useful as a higher-level clarification source.

Both types are subdivided into two ways of appliance:

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

2.1.3.1 Textual visualisation

Among the static textual visualisations are the feature map that can be printed to view the neural weights and the vector map that can 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.

2.1.3.2 Graphical visualisation

Static graphical visualisations have been embedded in the dynamic appliances. An anchored view is connected to this process when the user requests a snapshot during a process. The

---

9 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 takes about 8 minutes on a 66 Megahertz-486DX2 PC, 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.

10 See Appendix 2.1.3.x for textual and graphical visualisation examples.
evolving relations are shown for as long as the user does not explicitly request disconnection. A request for a snapshot after a process can be considered as the final state in that process.

During and after the training process the state of self-organisation for each dimension tuple can be examined. Here, the relations between neighbouring neurons are visualised as co-ordinates in the vector space. The interneuronal distances can also be visualised. Then, the Euclidean distances between neighbouring neurons are presented as thickness degrees in a static grid. 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 neuron. The second distribution perspective is of an exceptional nature, because it offers no clarification during the training process. The visualisation of the distribution of the best objects in the query requires a statistical frequency analysis in the pre-processing phase. This results in an ideal object distribution that can be useful in a comparison with the actual object distribution. The actual object distribution visualisation includes a search mechanism where any object can be entered and its best matching unit is returned to optimise comparisons.

Long term relations can be examined by visualising the recording objects. The error recording 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 object 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.

2.1.4 Accessibility

Accessibility, as defined in 2.1, stresses the importance of storage and retrieval of all input and output to enable reconstructions and variations. Therefore, the prototype supports three document-view formats that are invoked by an 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 that supports system- and user drawing as well as dynamically sizeable text. This format is used for graphical visualisations.

11 A dynamic b-tree has been used in the implementation to minimise 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.
FILTER Hitlist (*.hit), i.e. a special database format that uses Open Database Connectivity\textsuperscript{12} to connect only to external DBase (*.dbf) databases with the prototype database structure.

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

FILTER Settings (*.set), i.e. a special protected format that contains the configuration of parameters, preferences, files and paths and system settings.

FILTER Demo (*.dem), i.e. an extension of the settings format that 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 that 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 that 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 the user to optionally adjust the amount of returned information after the extraction process by merely resetting the view threshold.

2.2 Overview

To conclude this section a global overview of the prototype will be given in the form of an imaginary session\textsuperscript{13}.

A workspace appears when the session is started. It contains eight menus, a toolbar, a status bar and a system log-file\textsuperscript{14}. This log-file reports that the most recently used settings file has already been reloaded.

Only the corresponding data has to be loaded if this session is to be a continuation of the last session. 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.

\textsuperscript{12} Open Database Connectivity (ODBC) is Microsoft's standard interface that allows Windows applications to access many different types of databases.

\textsuperscript{13} See Appendix 2.2 for a detailed overview and explanations of FILTER’s workspace, the toolbar, various dialogue boxes and all menus with their statusbar descriptions.

\textsuperscript{14} The system log-file is named REPORT.TXT (See the print of FILTER’s workspace in Appendix 2.2).
However, the complete process cycle has to be pursued if this session is not to be a
continuation of the last session or if there is no data file that corresponds with these settings.
To start with, the settings most corresponding 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. These
new settings ought to be saved after fine-tuning. Optionally, the settings can be printed in a
listing format for referencing purposes.

Once 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. A number of new windows appear, depending on what visualisation preferences have
been set, that offer views from different perspectives onto the training process. These
visualisation preferences, like all settings, can be altered at any moment. For example, it is
possible to interfere by adjusting training parameters like the learn rate and the region update
area if the representation process does not seem to work out well. Of course, the process can
also be interrupted.

The system asks whether a table map for this feature map must be generated when the training
process ends. This question should be answered affirmative if a low context size has been
used. It is necessary to save the data after the table map generation 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. Three more new windows appear when the extract
process is started: 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 inactivated. 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 that is contained within the hitlist view looks
interesting, the Retrieve button in the view can be clicked 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. The view threshold can be adjusted to fine-tune retrieval and 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 executed to update the accessible subset of the database.

The hitlist ought to be saved once the extraction process has ended 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 since the prototype also supports smooth background processing.

2.3 Summary

Flexibility, performance, visualisation and accessibility were defined as essential application properties to create a well functioning neural filtering environment. Therefore, these application properties were investigated in depth. This led to the development of an event-driven, multitasking neural filtering environment with features like the table map, dynamic hitlists, extensive visualisation options and full implementation of object accessibility.

An imaginary session was described to give a global overview of the prototype.
Evaluation

3.0 Introduction

This chapter explains how the prototype, as described in chapter two, has been evaluated. It has been divided into three main sections.

The first section describes the preparation phase. 3.1.1 describes how the data set was composed. 3.1.2 explains how the queries were selected. 3.1.3 describes how the settings were varied. 3.1.4 defines correlation and other related terms. 3.1.5 describes how the correlation between the query representation in the feature map and each document in the data set was calculated. 3.2 anticipates some expected results.

The second section describes the execution and analysis phase. 3.3.1 provides a detailed report on the preliminary outcomes that are analysed in 3.3.2. Based on these preliminary outcomes additional tests and analyses have been carried out. These are described in detail in 3.3.3 and 3.3.4.

The third section describes a comparison made between FILTER and ZyIMAGE, a contemporary index-based information retrieval system. This chapter is concluded with a summary in 3.5.

3.1 Preparation

3.1.1 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 composed¹ by querying ZyIMAGE, an index-based document retrieval application. Three atomic queries were used: WAR*, EC and COMPUT*. The inclusion of wildcards 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 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”.

¹ See Appendix 3.1.1 for a complete oversight of the data set composition.
3.1.2 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.

The primary objective of this evaluation has been to investigate a wide range of parameter configurations. Although this implicated that only one query for each type could be evaluated within the available time-span, it must be taken into account that robustness is an essential property of artificial neural networks. Therefore, there does not seem to be much reason to believe that different queries would have yielded entirely different outcomes, although the results would become more dependable if more queries would have been evaluated.

3.1.2.1 Full-text query

The highest ranked document of the C-group was chosen as the full-text query\(^2\). 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\(^3\), are represented in the map after the training process.

Note that inclusion of the query document in the data set would influence the precision and recall ratios in a positive way. Therefore, the query document has been excluded from the precision and recall charts\(^4\) to present the results more impartially.

3.1.2.2 Artificial query

The artificial query was composed by concatenating all informative words in the full-text query separated by a comma\(^5\). 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 a better representation can be formed on the feature map after the training process because of its noiselessness. A practical advantage is the smaller size of the neural net needed.

---

\(^2\) See Appendix 3.1.2.1 for a print of the full-text query.

\(^3\) In the case of this query, in combination with trigrams, examples of features are COM, OMP, MPU, etc.

\(^4\) See Appendix 3.3.1 and 3.3.3 for the precision and recall charts.

\(^5\) See Appendix 3.1.2.2 for a print of the artificial query.
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 easier due to the artificial query’s resemblance to a weighted quorum query. Then, repetitions of keywords in the artificial query are translated into weights in the quorum query. This seems the only way to obtain a query that can be processed properly by both retrieval systems. On the one hand, an index-based retrieval system cannot process natural language consisting out of multiple words without inclusion of operators. On the other hand, a neural network-based retrieval system cannot process Boolean queries properly because they consist out of too few patterns to derive a just distributed representation. An artificial neural network simply needs more data before its properties begin to apply.

### 3.1.3 Settings

Four parameters have been investigated thoroughly:

- 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 that 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 neuron 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, the hit threshold has been kept constant at a value of 0.2 in the preliminary evaluation. 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 depth 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\(^6\) based on experimental pre-processing as well as on former research by others.

### 3.1.4 Definitions

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 maximum recall, one of which being the full-text query itself. These documents only are directly about computer manufacturers and about one of their computer models. Ten other documents were allowed to be retrieved to continue maximum precision. These documents are about computer-related companies or about computer-related products. All other 86 documents were rated irrelevant\(^7\).

The term *information value* has been used as the reciprocal of a pattern’s probability of occurrence\(^8\). This means that if the probability a pattern will occur is high, its information value is low.

The term *precision* indicates the number of correlating documents that are retrieved relative to the total number of retrieved documents.

The term *recall* indicates the number of retrieved correlating documents relative to the total number of correlating documents in the data set.

The term *accuracy* has been used to indicate the precision at 100% recall for settings that did not return any irrelevant documents before all extremely correlated documents were retrieved\(^9\).

### 3.1.5 Correlation calculation

Three alternative selection criteria have been calculated for each document in the data set to measure its correlation with the query. The system determines whether or not a document is to be retrieved based on one of these three values:

- The average error, i.e. the cumulation of the 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 because correlation is a result of distance

---

\(^6\) See Appendix 3.1.3 for a print of a complete setting used in this evaluation.

\(^7\) See Appendix 3.1.1 for correlation assignments.

\(^8\) This means that \(I = 1/P\), where \(I\) is the information value and \(P\) is the pattern’s probability of occurrence.

\(^9\) Since the entire data set has been examined manually beforehand it is known when all extremely correlated documents are returned.
calculations in this concept. Its value is composed out of information of all \textit{n}-grams, which results in a global measurement of a document.

The perfect hit rate, i.e. the cumulation of the number of \textit{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 \textit{n}-grams in the document. This can be thought of as a positive filter because correlation is a result of counting hits in this concept. Its value is not composed out of information of all \textit{n}-grams, which can result in a global as well as a local measurement of a document. This means that a document can still be retrieved if only a section in that document correlates to the feature map representation.

The average hit error, i.e. the cumulation of the Euclidean distance to the best matching unit in the neural net for all \textit{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 \textit{n}-grams in the document. This can be thought of as a positive-negative filter because 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 \textit{n}-grams, which can result in a global as well as a local measurement of a document. This means that this positive-negative filter could also clarify how well a correlating section of a document correlates to the feature map.

### 3.2 Expectations

After training, the representation of the full-text query in the feature map is 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\(^{10}\). Therefore, the full-text query should also be less accurate in the extraction process.

[artificial +, full-text -]

The query’s discriminating features will fade too much if the generalisation factor becomes too high. Then, no accurate correlation between the data flow and the query representation can be distinguished anymore.

[generalisation factor +, accuracy -]

The information value of a pattern increases exponentially with the context size. In other words, the patterns in the feature map as well the patterns of the data flow are more distinctive when the context size is set 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 that should at least do well in the case of the full-text query with inclusion of the space as a character.

[average error correlation +/-, perfect hit-sensitive correlation ++/--]

### 3.3 Execution

#### 3.3.1 Preliminary results

In this section the preliminary results table\(^{11}\) is globally reviewed, from left to right and from top to bottom.

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. The average error of the query was always higher than the hit threshold whenever extremely accurate hit threshold-sensitive results were returned.

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

---

\(^{10}\) See Appendix 3.2.x for prints of the internal data flow of the full-text query and the artificial query as well as prints of the common words and the common endings.

\(^{11}\) The preliminary results table is printed in full in Appendix 3.3.1.
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 accuracy decreased again. At this point the generalisation factor became relevant again.

When sorted on the average hit error the results were also extremely accurate in some configurations. In general though, they turned out to be less stable than the results sorted on 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.

### 3.3.2 Preliminary analysis

In this section the results, as described in 3.3.1, are interpreted.

The artificial query is noiseless. Because of this property any generalisation of its data will 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 a too 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 the space as a character the information value of the patterns in the full-text query can become too low. Also, the relative number of relevant patterns increases significantly by inclusion of the space as a character and thus the inclusion of adjacent word relations. Therefore, the information value increases as well as the accuracy of the query’s representation in the feature map with the inclusion of the space as a character.

The context size must not be set too high to avoid an inaccurate representation of discriminating features. However, the generalisation factor and the space as a character become relatively insignificant 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. Although the average

---

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

13 For the query used in this evaluation, discriminating features are for example DELL, CHIP, INTEL, etc.
error will be of a relatively indeterminate nature, the perfect hits these few patterns will cause are extremely informative when the hit threshold is set lower than the maximum map activity and the hitlist is sorted on the perfect hit rate.

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.

New settings have been evaluated to validate this analysis. There has been focused on configurations with a low context size to minimise execution times. The results will be reviewed in 3.3.3.

### 3.3.3 Additional results

In this section the additional results table\(^\text{14}\) is globally reviewed, from top to bottom.

To validate the assumption that the hit threshold should be set lower than the maximum map activity, one extremely accurate configuration\(^\text{15}\) 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\(^\text{16}\). Accuracy increased as the hit threshold was set lower. A state of extreme accuracy was reached when a hit threshold was used of 50 percent below the minimum map error.

At this point the experiment was transferred to the only accurate average error-sorted result\(^\text{17}\). 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

---

\(^{14}\) The additional results table is printed in full in Appendix 3.3.3.

\(^{15}\) Settings properties: full-text query, low generalisation factor, high context size, no space as character.

\(^{16}\) Settings properties: full-text query, low generalisation factor, low context size, no space as character.

\(^{17}\) Settings properties: full-text query, low generalisation 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. The results were not good at all when a hit threshold was used of 50 percent below the maximum map activity. However, the results became extremely accurate when a hit threshold of 25 percent was used.

This left only the artificial query to be optimised because accurate results with the artificial query would make a comparison with an index-based information retrieval system easier. First, an experiment with a configuration with no generalisation factor was carried out in detail. Again, the results became accurate when a hit threshold of 50 percent below the maximum map activity was used. 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. Again, the results became accurate when a hit threshold of 50 percent below the maximum map activity was used. However, the results became highly inaccurate again when a value of 75 percent below the maximum map activity was used.

### 3.3.4 Additional analysis

The hit threshold is the most essential parameter. A value of 50 percent below the maximum map activity generally 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.

Robustness and execution time increase with the context size. 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.

The artificial query performs best when it is not compressed.

---

18 Settings properties: full-text query, medium generalisation factor, low context size, space as character.

19 Settings properties: artificial query, no generalisation factor, low context size.

20 Settings properties: artificial query, low generalisation factor, low context size.
3.4 Comparison

The data set was also evaluated with ZyIMAGE, a contemporary index-based information retrieval system. In ZyIMAGE a query consists of one or more keywords. Multiple keywords are controlled by a set of logical and statistical operators.

However, it seems not that clear 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 much more 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. A more valuable comparison could have been made if the artificial query had yielded the best results. Then, the same query could have been used in both systems.

Having emphasised the relative importance of any comparison made between these systems, the results obtained with ZyIMAGE are reviewed\(^2\) 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 atomic 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 atomic 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. At 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.

\(2\) The ZyIMAGE results tables are printed in full in Appendix 3.4.
Inclusion of fuzziness in the searches caused a decrease in accuracy in almost all cases. This is probably caused by the rather good quality of the scanned articles. Autonomously generated pattern permutations will only decrease accuracy if most matches can already be found without fuzziness. In such a case, most of the fuzzy matches will probably be unintended and will cause a retrieval overkill.

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.

### 3.5 Summary

The data set with which the prototype was evaluated is a small subset of a corpus of recently scanned newspaper articles. A full-text query and an artificial query were taught to the neural net. The settings were configurations with varying generalisation factors, context sizes, spaces as characters and, although only after the preliminary results, hit thresholds. Three levels of query correlation were determined manually for all data documents. The terms information value, precision, recall and accuracy were also defined. To determine the correlation between the query representation in the feature map and each document in the data set a negative filter, a positive filter and a positive-negative filter have been evaluated.

Expected was that the artificial query would yield better results than the full-text query, that an increase of the generalisation factor would decrease accuracy and that a high context size and the inclusion of the space as a character would increase accuracy.

In the preliminary results the artificial query didn’t do well at all. The full-text query did much better. The results were best when the space as a character was used. The perfect hit rate turned out to be the most promising correlation calculation in combination with a high context size. Then the results became extremely accurate. Therefore, the hit threshold was examined in detail in the additional results. Also the artificial query was evaluated again, but this time with no generalisation factor.

In the additional results the hit threshold turned out to be the most important parameter of all. All results could be optimised so that no irrelevant documents were returned anymore. As a
rule of thumb, its value should be set at 50 percent below the maximum map activity. Perfect retrieval was accomplished with a configuration containing the full-text query, a low generalisation factor, a low context size, with the space as a character and a hit threshold set at 50 percent below the maximum map activity. The hitlist was sorted on the perfect hit rate.

A comparison was made between FILTER and ZyIMAGE after emphasising the relative importance of any comparison made between a neural and a index-based information retrieval system. In this comparison with this data set and this query, the index-based information retrieval system using Boolean queries could also achieve perfect retrieval. The artificial query seemed to contain too much information for the index-based system. In general, although these results were identical, the neural filter seems more flexible and more robust.
Discussion

4.0 Introduction

This chapter concludes the main section of this report. It has been divided into two main sections.

The first section recapitulates the main consequences that can be drawn from the evaluation. In the second section some directions for future research are proposed. This chapter is concluded with the bottom line in 2.3.

4.1 Conclusions

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\(^1\).

4.2 Future research

4.2.1 Corpus representation

Although the neural filter is likely to exhibit more flexible and more robust behaviour than an 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

\(^1\) 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 in a 16-bit environment, while maintaining accurate retrieval.
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 that still contains all data patterns to maintain maximum performance and minimise data storage overhead. All pattern repetitions can be eliminated by adding the number of occurrences in the text part to each data pattern. This way the character-to-vector translations and all repetitional cycles are eliminated from the corpus extraction process. Although these measures seriously affect the flexibility of the 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.

4.2.2 Data fusion

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.

4.3 Summary

This report indicates that the neural filter could contribute significantly to the class of real-time filtering applications as a high quality full-text search method.
The references listed here represent only the intensively used literature sources. For an extensive listing of references see [Scholtes, 1994].


Appendices

This extensive appendices section provides a more profound insight in the FILTER project through additional clarification and background information. It uses parts of the hierarchical structure of the main section to provide implicit cross-referencing.
Background [Appendices]

1.3 Neural filter

Principle
Reprinted from [Scholtes 1993]:

\[ n \text{-gram analysis method} \]
Reprinted from [Scholtes 1993]:

\[ \text{Neural Map holding character n-grams} \]
\[ \text{Input Fibres with weight } w(t) \text{ and input activation } x(t) \]
\[ \text{Shifting Window holding n elements (characters)} \]
Algorithm (schematic version)

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 these 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 these 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
Prototype [Appendices]

2.0 Introduction

In all examples given here to clarify various aspects of the prototype, the same data set has been used (FTQ23106.DAT). It is this data set which yields perfect retrieval in the evaluation, as is shown in the evaluation section.

2.1.1 Flexibility

The declaration of the feature map object is given here as a source code example to illustrate the concept of object-oriented programming. DECLARE_DYNCREATE is a Windows macro that provides automatic registration of the object when its constructor is called [i.e. CFeatureMap()]. Curly brackets are used as implicit inline-declaration markers. This class is application-independent; all its properties and capabilities are defined within itself.

class CFeatureMap : public CObject
{
    public:
        DECLARE_DYNCREATE(CFeatureMap)
        CFeatureMap();
        CFeatureMap(UINT x, UINT y, UINT z, UINT nRandomSpread = 15);
        ~CFeatureMap();

        // The embedded feature map object which contains the neuron objects
        CObArray map;

        // Library objects for communication with the interface
        CLib lib;
        CReport report;

        // Object vars (to speed up the most frequently used function members)
        float m_fMinDistance;
        float m_fCurrentDistance;
        float m_fTotalDistance;

        // Operations
        void Add(CFeatureMap* pDimension) {map.Add(pDimension);} 
        void Add(CFloatArray* pSensor) {map.Add((CObject*)pSensor);}

}
void Convert(CVectorArray* pFromVector, CFloatArray* pToVector);
void CopyRegion(CFloatArray* pFromNeuron, UINT nFromPosition, UINT nToPosition,
        CFloatArray* pToNeuron);
CFeatureMap* GetAt(int i) {return (CFeatureMap*)map.GetAt(i);}  
CString GetBMO(CVectorMap* pVectorMap, CFloatArray* pNeuron,UINT nConcatenations);
CPoint GetBMU(UINT nMaxX, UINT nMaxY, CFloatArray* pCurrentPattern);
CPoint GetBMU(UINT nMaxX, UINT nMaxY, CFloatArray* pCurrentPattern, long lCurrentCycle, UINT &nCurrentErrorPosition, UINT ErrorCompression, UINT nMaxError, float &fPeekDistance, __CFloatArray* pErrorHistory = NULL);
// GetDistance() is declared explicitly inline
float GetDistance(CFloatArray* pSensor, CFloatArray* pCurrentPattern);
CFloatArray* GetNeuron(UINT x,UINT y) {return GetAt(x) - GetSensor(y);}
CFloatArray* GetSensor(int i) {return (CFloatArray*)GetAt(i);}
int GetSize() {return map.GetSize();}
BOOL Load(CStdioFile &file, UINT &nX, UINT &nY, UINT &nSensors, UINT &nRegions,
        UINT &nRandomSpread);
float MinimalError(UINT nMaxX, UINT nMaxY, CFloatArray* pCurrentPattern, long lCurrentCycle, UINT &nCurrentActivityPosition, UINT nErrorCompression, UINT nMaxError, float &fPeekActivity, __CFloatArray* pActivityHistory, BOOL bRecordError);
float MinimalErrorDebug(UINT nMaxX, UINT nMaxY, CFloatArray* pCurrentPattern,
        long lCurrentCycle, UINT &nCurrentActivityPosition, UINT nErrorCompression, UINT nMaxError, float &fPeekActivity, __CFloatArray* pActivityHistory, BOOL bRecordError, CPoint &BMUcoordinate);
void Print();
void Reconstruct(UINT nX,UINT nY,UINT nZ,CFloatArray* pWeights);
void RecordError(float fError, long lCurrentCycle, UINT &nCurrentErrorPosition, UINT nErrorCompression, UINT nMaxError,
        __CFloatArray* pErrorHistory, float &fPeekDistance, BOOL bPeekIsMaximum = TRUE);
BOOL Save(CStdioFile &file, UINT nX, UINT nY, UINT nSensors, UINT nRegions, UINT nRandomSpread);
void SetAt(UINT i, CFloatArray* pSensor) {map.SetAt(i,(CObject*)pSensor);}  
void SetNeuron(UINT x, UINT y, CFloatArray* pSensor);
2.1.2.1 Dynamic k-d tree

Visualisation
Feature map elements as leaf nodes in a tree structure - Reprinted from [Koikkalainen 1990]:

![Dynamic k-d tree visualisation](image)

2.1.3.1 Textual visualisation

Feature map
This is a fragment of the contents of a feature map. Each neuron 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 neurons, which means that this region will recognise patterns that have a character in the middle that maps to 1.000000 (see vector map).

---

FEATURE MAP initialised with current settings...
- X-dimension : 13
- Y-dimension : 17
- Context size : 3
- Sensors/Neuron : 9
- Random spread : 15
neuron[0,0] : 0.224287 0.952296 0.029184 0.029922 0.687756 0.486086 0.873835 0.900448 0.889078
neuron[0,1] : 0.127397 0.976929 0.028571 0.372500 0.536198 0.871545 0.803125 0.985772 0.996528
neuron[0,2] : 0.303389 0.937831 0.002000 0.496275 0.509724 0.998167 0.492377 0.992838 0.805812
neuron[0,3] : 0.377346 0.997539 0.003503 0.542813 0.995419 0.999993 0.038903 0.779591 0.996528
neuron[0,4] : 0.697131 0.958209 0.025608 0.967122 0.999992 0.999998 0.091885 0.198423 0.993085
neuron[0,5] : 0.558126 0.403755 0.226493 0.997551 0.999998 0.999998 0.006682 0.016472 0.949989
neuron[0,6] : 0.974968 0.136078 0.503579 0.994403 0.999998 0.999998 0.125672 0.570563 0.768181
neuron[0,7] : 0.697058 0.043202 0.876233 0.807937 0.999998 0.999998 0.040433 0.202237 0.363312
neuron[0,8] : 0.343057 0.355185 0.932372 0.756739 0.999777 0.999998 0.808857 0.021074 0.368147
neuron[0,9] : 0.096726 0.503642 0.512291 0.715875 0.999998 0.999998 0.999998 0.076557 0.223334
neuron[1,0] : 0.502821 0.997213 0.978075 0.999998 0.999998 0.449903 0.631282 0.184550
neuron[1,1] : 0.685060 0.869337 0.995705 0.894068 0.902305 0.999998 0.661747 0.613795 0.036563
neuron[1,2] : 0.798250 0.979789 0.998584 0.509939 0.576002 0.999993 0.676429 0.866105 0.002862
neuron[1,3] : 0.922398 0.594168 0.717885 0.372101 0.523254 0.831094 0.591289 0.945196 0.592083
neuron[1,4] : 0.988563 0.380237 0.419349 0.022776 0.403899 0.420415 0.513318 0.999918 0.999917
neuron[1,5] : 0.001105 0.286195 0.012265 0.066698 0.544840 0.558795 0.600750 0.999992 0.999998
neuron[1,6] : 0.242177 0.625855 0.311376 0.016595 0.297078 0.142991 0.847766 0.988551 0.100470
neuron[1,7] : 0.808413 0.956070 0.631514 0.031282 0.485015 0.505438 0.518370 0.998363 0.836606
neuron[1,8] : 0.295684 0.815231 0.462251 0.262446 0.609304 0.773652 0.245971 0.922429 0.826050

Vector map
The vector map defines the mapping for each data pattern into its internal vector representation:

VECTOR MAP initialised with current settings...

<table>
<thead>
<tr>
<th>Code spread</th>
<th>Code spread</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.000000</td>
<td>0.000000</td>
</tr>
<tr>
<td>0.500000</td>
<td>0.500000</td>
</tr>
<tr>
<td>1.000000</td>
<td>1.000000</td>
</tr>
</tbody>
</table>

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
Examples of snapshots of dynamic textual visualisations are printed in Appendix 3.2.x for the full-text query and the artificial query that have been used in the evaluation of the prototype.

**Object frequency analysis**

This is a fragment of the frequency analysis of the full-text query used in the evaluation of the prototype. It is used in the visualisation of the best query objects in the feature map after the training process:

```
PRINTING ALL 449 OBJECTS IN QUERY...

_CO  11  LL_  7  INE  6
ENT  9  MAC  7  MPU  6
ER_  9  NTI  7  POR  6
_MA  7  OMP  7  PUT  6
_PE  7  ORP  7  UTE  6
CHI  7  PEN  7  _PR  5
COM  7  TER  7  DEL  5
COR  7  TIU  7  [...]  
IUM  7  UM_  7  
```


2.1.3.2 Graphical visualisation

**State of self-organisation**

A perfect state of self-organisation could have been reached if the data would have been distributed homogeneously. Then all neurons would have been equally distant from their neighbours. In the visualisation this would have resulted in a perfect grid. However, patterns in natural language are not homogeneously distributed. Therefore, the visualisation here printed is actually quite good because the neurons are using quite a lot of the vector space.
Interneuronal distances

This fragment shows the feature map from another, more static perspective where cluster boundaries can easily be traced.
Object distribution (best objects possible)

A comparison with the textual visualisation of the feature map shows that the cluster observed there is mapped to the space character due to the fourth vector dimensions. (In the prototype the space character is represented by an underscore character.)
Object distribution (best objects in query)

The empty co-ordinates are due to the fact that some patterns have the same neuron as their BMU. Only the most frequent object is printed in that case.
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 10 times to the feature map. The overall errors get smaller as the feature map converges to its final representation. This means that the feature map finds an overall way to represent the query. Peaks are due to infrequent patterns. A picture compression of $n$ means that only one per $n$ errors is plotted to give an overall impression.
**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.
2.2 Overview

The workspace, the toolbar, a selection of dialogue boxes and all menus and their statusbar descriptions are presented here to give a general impression of the prototype.

FILTER workspace

In this snapshot all system documents have already been opened. FTQ23106.HIT is an additional document to exemplify the hitlist system. At this point new data, including the table map, has been loaded. The Extract Data toolbar button is having focus. This means that the cursor, although invisible on screen-dumps, is being slided over this button. This event causes an appearance of a small temporary window, just below the toolbar button, with context-sensitive information as well as a more detailed command-description in the status bar. This mechanism is known as Hover-help or Tooltips.

FILTER toolbar, after loading data, off line

In this snapshot all system documents have already been opened. FTQ23106.HIT is an additional document to exemplify the hitlist system. At this point new data, including the table map, has been loaded. The Extract Data toolbar button is having focus. This means that the cursor, although invisible on screen-dumps, is being slided over this button. This event causes an appearance of a small temporary window, just below the toolbar button, with context-sensitive information as well as a more detailed command-description in the status bar. This mechanism is known as Hover-help or Tooltips.
FILTER toolbar, during the extraction process

The toolbar contains the most frequently used commands:

- Items 1 and 2 represent respectively File-Open and File-Save As.
- Items 3, 5, 6 and 7 represent respectively Settings-Load Settings, Settings-Parameters, Settings-Preferences and Settings-Files and Paths.
- Items 4, 8, 9, 10 and 19 represent respectively Process-Load Data, Process-Initialise Objects, Process-Teach Query, Process-Extract Text Parts and Process-Interrupt.
- Items 11, 12, 13, 14 and 15 represent respectively Hitlist-Sort Hitlist, Hitlist-First Record, Hitlist-Previous Record, Hitlist-Next Record and Hitlist-Last Record.
- Item 16 represents View-Redraw Toolbar.
- Items 17 and 18 represent respectively Help-About... and Help-Context Help.

Command Settings-Parameters

The parameters dialogue box contains primarily integer values:

- Context size: the size of the window that is being shifted over the data.
- Sensors: the number of weights, or dimensions, of one object vector.
- X-neurons: X-dimension of the feature map.
Value spread: the initial sensor value range.

Objects: the number of objects / characters.

Value spread: the object-vector value range.

Capacity: maximum number of recordable errors.

Compression: the error recording sampling rate.

Cycles: the number of cycles a pattern from the query is presented in the training process.

Sigma start: the initial region size of the feature map that is being updated after a training cycle.

Sigma end: the final region size; the map is forced to converge when this parameter is set smaller than sigma start.

Epsilon start: the initial learning rate, i.e. the initial measure of adjustment of the neuron weights.

Epsilon end: the final learning rate; the map is forced to stabilise when this parameter is set smaller than epsilon start.

Hit threshold: the correlation ratio a pattern must have to its BMU to be recorded as a perfect hit.

View threshold: the correlation ratio a text part must have with the query to be visible in the hitlist.

Text part separator string: the text part separator within a document of the data flow.

Cycle interval: the visualisation update interval in cycles.

Picture size: the dimensions of the visualisations in internal units.

Accentuation: the distinctiveness degree of the interneuronal distances in internal units.

From dimension: the first dimension in the visualisation of the map’s state of self-organisation.

To dimension: the last dimension in the visualisation of the map’s state of self-organisation.

Minimum length: the minimum length a word must have to not be flushed during input filtering.

Hints Button: opens the Hints Dialogue where important parameters can be calculated automatically.

Demo Button: opens the Demo Dialogue where special demo parameters can be set.

“Arrow” Buttons: increase or decrease the value in the edit control.
The preferences dialogue box contains Boolean values:

- **Activate Files & Paths**: enable access to the Files and Paths dialogue box.
- **Activate System**: enable access to the system menu.
- **Constant**: inactivate the Epsilon end parameter.
- **Decreasing in time**: use the Epsilon start parameter as well as the Epsilon end parameter.
- **Use text part separator**: enable the text part separator to also determine text parts within a document.
- **Use feature map**: present the data flow to the feature map during the extraction process.
- **Use table map**: present the data flow to the table map during the extraction process.
- **Space is a character**: convert all non-characters to one space instead of concatenating all characters.
- **Common words**: flush all common words from the input.
- **Common endings**: flush all common endings from the remaining input strings.
- **Small words**: flush all strings from the remaining input smaller than the minimum word length.
- **Pattern activity**: print the activity of each pattern in the data flow (for external spreadsheet visualisation).
- **Text part activity**: print final correlation calculations to the system report.
- **Internal data flow**: print the data as it is presented to the feature map or table map.
- **Tile picture vertically**: order the document views at the beginning of the training process.
- **Self-organisation**: visualise the current state of self-organisation of the feature map.
- **Interneuronal distances**: visualise the distances between neighbouring neurons in the feature map.
- **Object distribution**: visualise the best matching object for each neuron in the feature map.
- **Error recording**: visualise the entire training process by plotting the recorded errors.
- **Activity recording**: visualise the degree of correlation of the data flow during the extraction process.
The Files and Paths dialogue box contains mainly string values:

- **Browse Buttons**: select a file or directory to replace in the edit control.
- **Edit Buttons**: open the document for examining or editing purposes.
- **Check Buttons**: implicitly use this file in the process or select once at run-time when needed.
- **Reset Button**: delete all files in the list box or undo deletion.
- **Delete Button**: delete the selected file in the list box.
- **Add Button**: add all files with the specified extension in the specified directory to the list box.
- **Extension Edit Control**: specify the file extension.
- **Dir Button**: specify the directory.
- **Count Button**: count the current number of files in the list box.
The Hints dialogue box contains mainly buttons:

Calculate generalisation factor: the number of query-objects divided by the number of neurons.

Calculate maximum vector spread: the minimum power of \( n \) to the number of sensors per neuron greater than or equal to the number of objects minus one. In this example \( n \) yields a vector dimension spread of \( 1/n \), i.e. for example \([0.0, 0.5, 1.0]\).

Calculate number of learn cycles, based on \( n \) parses: the number of training cycles needed to parse the query once, multiplied with \( n \) parses.

Calculate recording capacity: the dimension of the error recording object, which may not exceed a certain maximum for memory management purposes.

Calculate hit threshold, based on a deviation of \( n \) percent: the maximum map activity minus \( n \) percent.

Calculate All Button: calculate all hints at once.

Interrupt Button: interrupt the hint calculations.
Command Settings-Parameters-Demo

The Demo dialogue box contains special options:

- **Shuffle/Nr of shuffles**: shuffle the generated vector map \( n \) times.
- **Include extraction**: include the extraction process in the demo. This option can only be activated when a Settings file is run as a demo.
- **Feature map**: print the weights of all neurons in the feature map at the end of the training process.
- **Vector map**: print the vector map at the beginning of the demo.
- **Error recording**: print the contents of the error recording object at the end of the training process.
- **Activity recording**: print the contents of the activity recording object at the end of the demo.

Command Process-Table Map Menu

A table map is a two-dimensional hash table which can be used in the extraction process to dramatically speed up execution by calculating all possible patterns in advance. However, with the context size, the table map size increases exponentially. This can result in long generation times...

Needed map table entries for the current settings: 19683
Command Hitlist-Sort Hitlist

A hitlist can be sorted on: Text part (which results in a chronological order), Average error, Perfect hit rate, Average hit error and Filename.

Command Settings-Evaluation Mode

The Evaluate Settings dialogue box:

Add Button: select a settings file to add to the list box.
Delete Button: delete the selected file in the list box.
Reset Button: delete all files in the list box or undo deletion.
Count Button: count the current number of files in the list box.
Add Dir Button: add all settings files in the settings directory to the list box.
Evaluate Button: Start processing all settings in the list box automatically.
FILTER menus

<table>
<thead>
<tr>
<th>File</th>
<th>Settings</th>
</tr>
</thead>
<tbody>
<tr>
<td>New</td>
<td>Parameters</td>
</tr>
<tr>
<td>Open</td>
<td>Preferences</td>
</tr>
<tr>
<td>Close</td>
<td>Files and Paths</td>
</tr>
<tr>
<td>Save</td>
<td>Load Settings</td>
</tr>
<tr>
<td>Save As</td>
<td>Save Settings</td>
</tr>
<tr>
<td>Print</td>
<td>Print Settings</td>
</tr>
<tr>
<td>Print Preview</td>
<td>Evaluation Mode</td>
</tr>
<tr>
<td>Print Setup</td>
<td></td>
</tr>
</tbody>
</table>

Process

<table>
<thead>
<tr>
<th>Initialize Objects</th>
<th>Feature Map</th>
</tr>
</thead>
<tbody>
<tr>
<td>Teach Query</td>
<td>Vector Map</td>
</tr>
<tr>
<td>Extract Text Parts</td>
<td>Common Words</td>
</tr>
<tr>
<td></td>
<td>Common Endings</td>
</tr>
<tr>
<td>Load Data</td>
<td>Error Recording</td>
</tr>
<tr>
<td>Save Data</td>
<td>Activity Recording</td>
</tr>
<tr>
<td>Print</td>
<td>Best Objects in Query</td>
</tr>
<tr>
<td>Visualize</td>
<td>Best Objects Possible</td>
</tr>
<tr>
<td>Table Map Menu</td>
<td>Best Objects in Query</td>
</tr>
<tr>
<td>Interrupt</td>
<td>Self-organization</td>
</tr>
<tr>
<td>Busy</td>
<td>Interneuronal Distances</td>
</tr>
<tr>
<td></td>
<td>Object Distribution</td>
</tr>
<tr>
<td></td>
<td>Error Recording</td>
</tr>
<tr>
<td></td>
<td>Activity Recording</td>
</tr>
<tr>
<td>Output</td>
<td>Picture</td>
</tr>
<tr>
<td>---------------</td>
<td>------------------</td>
</tr>
<tr>
<td>Find</td>
<td>Ctrl+P</td>
</tr>
<tr>
<td>Replace</td>
<td>Ctrl+R</td>
</tr>
<tr>
<td>Repeat Last Find</td>
<td>Ctrl+L</td>
</tr>
<tr>
<td>Undo</td>
<td>Ctrl+Z</td>
</tr>
<tr>
<td>Erase</td>
<td>Ctrl+E</td>
</tr>
<tr>
<td>Cut</td>
<td>Ctrl+X</td>
</tr>
<tr>
<td>Copy</td>
<td>Ctrl+C</td>
</tr>
<tr>
<td>Paste</td>
<td>Ctrl+V</td>
</tr>
<tr>
<td>Read Only</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Picture</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Clear All</td>
<td>Mouse Drawing</td>
</tr>
<tr>
<td></td>
<td>Thick Pen</td>
</tr>
<tr>
<td>Pen Widths</td>
<td>Colors</td>
</tr>
</tbody>
</table>

### Hitlist

<table>
<thead>
<tr>
<th></th>
<th>F8</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sort Hitlist</td>
<td>Print Hitlist in ASCII format</td>
</tr>
<tr>
<td>Search Text Part</td>
<td>Retrieve Text Part</td>
</tr>
<tr>
<td>Change Retrieval Paths</td>
<td></td>
</tr>
<tr>
<td>First Record</td>
<td>Ctrl+PageUp</td>
</tr>
<tr>
<td>Previous Record</td>
<td>PageUp</td>
</tr>
<tr>
<td>Next Record</td>
<td>PageDown</td>
</tr>
<tr>
<td>Last Record</td>
<td>Ctrl+PageDown</td>
</tr>
</tbody>
</table>

### Delete Record

- Delete All Records

### View

- Toolbar
- Status Bar
- Redraw Toolbar: Alt+R

### Window

- Cascade
- Tile Horizontally
- Tile Vertically
- Arrange Icons

### System

- Setup Options
- Update Common Words
- Update Common Endings
- Destroy All Data Objects
- Compare Extraction Methods

### Help

- Index: F1
- Context Help: Shift+F1
- Using Help
- System Info
- Run Demo: F2
- About...

Note: only one of the three menus Output, Picture and Hitlist is shown. This depends on the type of the active view.
FILTER statusbar

The process indicator is switched on as the prototype starts processing. This results in an accessibility-update for all commands. Also progress is send to the status bar in percentages and is displayed. For example, during the extraction process the status bar displays Processed a % of file b at text part c, file position d. However, if an accessible command gets focus, i.e. if the user slides the cursor over a toolbar button, presses the left mouse button on a toolbar or activates a menu item, the statusbar provides an explanation for that command.

<table>
<thead>
<tr>
<th>Menu command</th>
<th>Statusbar description</th>
</tr>
</thead>
<tbody>
<tr>
<td>File-New</td>
<td>Create a new document</td>
</tr>
<tr>
<td>File-Open</td>
<td>Open an existing document</td>
</tr>
<tr>
<td>File-Close</td>
<td>Close the active document</td>
</tr>
<tr>
<td>File-Save</td>
<td>Save the active document</td>
</tr>
<tr>
<td>File-Save As</td>
<td>Save the active document with a new name</td>
</tr>
<tr>
<td>File-Print</td>
<td>Print the active document</td>
</tr>
<tr>
<td>File-Print Preview</td>
<td>Display full pages</td>
</tr>
<tr>
<td>File-Print Setup</td>
<td>Change the printer and printing options</td>
</tr>
<tr>
<td>File-4 MRU Files</td>
<td>Open this document</td>
</tr>
<tr>
<td>File-Exit</td>
<td>Quit the application; prompts to save documents</td>
</tr>
<tr>
<td>Settings-Parameters</td>
<td>Open the parameters dialogue to edit process parameters</td>
</tr>
<tr>
<td>Settings-Preferences</td>
<td>Open the preferences dialogue to edit process preferences</td>
</tr>
<tr>
<td>Settings-Files and Paths</td>
<td>Open the files and paths dialogue to edit process files and paths</td>
</tr>
<tr>
<td>Settings-Load Settings</td>
<td>Load a configuration of parameters, preferences and files &amp; paths</td>
</tr>
<tr>
<td>Settings-Save Settings</td>
<td>Save the current configuration of parameters, preferences and files &amp; paths</td>
</tr>
<tr>
<td>Settings-Print Settings</td>
<td>Change the printer and printing options</td>
</tr>
<tr>
<td>Settings-Evaluation Mode</td>
<td>Select a set of settings to evaluate completely unsupervised</td>
</tr>
<tr>
<td>Process-Initialise Objects</td>
<td>Initialise all data objects</td>
</tr>
<tr>
<td>Process-Teach Query</td>
<td>Teach the query to the feature map object</td>
</tr>
<tr>
<td>Process-Extract Text Parts</td>
<td>Extract text parts, correlating with the query</td>
</tr>
<tr>
<td>Process-Load Data</td>
<td>Load a configuration of objects</td>
</tr>
<tr>
<td>Process-Save Data</td>
<td>Save the current configuration of objects</td>
</tr>
<tr>
<td>Process-Print-Feature Map</td>
<td>Print the current feature map object</td>
</tr>
<tr>
<td>Process-Print-Vector Map</td>
<td>Print the current vector map object</td>
</tr>
<tr>
<td>Process-Print-Common Words</td>
<td>Print the current common words object</td>
</tr>
<tr>
<td>Process-Print-Common Endings</td>
<td>Print the current common endings object</td>
</tr>
<tr>
<td>Process-Print-Error Recording</td>
<td>Print the error recording object</td>
</tr>
<tr>
<td>Process-Print-Activity Recording</td>
<td>Print the activity recording object</td>
</tr>
<tr>
<td>Process-Print-Best Objects in Query</td>
<td>Print all objects in the query, sorted on descending frequency</td>
</tr>
<tr>
<td>Process-Visualise-Self Organisation</td>
<td>Visualise the state of self-organisation of the feature map object</td>
</tr>
<tr>
<td>Process-Visualise-Interneuronal Distances</td>
<td>Visualise the interneuronal distances in the feature map</td>
</tr>
<tr>
<td>Process-Visualise-Object Distribution-Best Objects Possible</td>
<td>Visualise the object distribution by generating objects</td>
</tr>
<tr>
<td>Process-Visualise-Object Distribution-Best Objects in Query</td>
<td>Visualise the object distribution by matching the query objects in the query</td>
</tr>
<tr>
<td>Process-Visualise-Error Recording</td>
<td>Visualise the error recording object</td>
</tr>
<tr>
<td>Process-Visualise-Activity Recording</td>
<td>Visualise the activity recording object</td>
</tr>
<tr>
<td>Process-Interrupt</td>
<td>Interrupt the current process</td>
</tr>
<tr>
<td>Process-Busy</td>
<td>Override deactivation of the process indicator switch</td>
</tr>
<tr>
<td>Menu command</td>
<td>Statusbar description</td>
</tr>
<tr>
<td>-----------------------</td>
<td>------------------------------------------------------------</td>
</tr>
<tr>
<td>Output-Find</td>
<td>Find the specified text</td>
</tr>
<tr>
<td>Output-Replace</td>
<td>Replace specific text with different text</td>
</tr>
<tr>
<td>Output-Repeat Last</td>
<td>Repeat the last action</td>
</tr>
<tr>
<td>Output-Undo</td>
<td>Undo the last action</td>
</tr>
<tr>
<td>Output-Erase</td>
<td>Erase the selection</td>
</tr>
<tr>
<td>Output-Cut</td>
<td>Cut the selection and put it on the Clipboard</td>
</tr>
<tr>
<td>Output-Copy</td>
<td>Copy the selection and put it on the Clipboard</td>
</tr>
<tr>
<td>Output-Paste</td>
<td>Insert Clipboard contents</td>
</tr>
<tr>
<td>Output-Read Only</td>
<td>Toggle the activation of keyboard input</td>
</tr>
<tr>
<td>Picture-Clear All</td>
<td>Erase everything</td>
</tr>
<tr>
<td>Picture-Mouse Drawing</td>
<td>Toggle the activation of user drawing with mouse</td>
</tr>
<tr>
<td>Picture-Thick Pen</td>
<td>Toggle between thick and thin pen</td>
</tr>
<tr>
<td>Picture-Pen Width</td>
<td>Open the dialogbox where the Pen sizes can be defined</td>
</tr>
<tr>
<td>Picture-Colors</td>
<td>Choose a different painting color</td>
</tr>
<tr>
<td>Hitlist-Sort Hitlist</td>
<td>Sort the hitlist</td>
</tr>
<tr>
<td>Hitlist-Print Hitlist in ASCII format</td>
<td>Convert the hitlist from DBF to ASCII and show the result</td>
</tr>
<tr>
<td>Hitlist-Search Text Part</td>
<td>Search for a text part</td>
</tr>
<tr>
<td>Hitlist-Retrieve Text Part</td>
<td>Retrieve this text part</td>
</tr>
<tr>
<td>Hitlist-Change Retrieval Paths</td>
<td>Change all the retrieval paths of the hitlist file</td>
</tr>
<tr>
<td>Hitlist-First Record</td>
<td>Move to the first record in the hitlist</td>
</tr>
<tr>
<td>Hitlist-Previous Record</td>
<td>Move to the previous record in the hitlist</td>
</tr>
<tr>
<td>Hitlist-Next Record</td>
<td>Move to the next record in the hitlist</td>
</tr>
<tr>
<td>Hitlist-Last Record</td>
<td>Move to the last record in the hitlist</td>
</tr>
<tr>
<td>Hitlist-Delete Record</td>
<td>Delete a hit from the hitlist</td>
</tr>
<tr>
<td>Hitlist-Delete All Records</td>
<td>Zap the hitlist</td>
</tr>
<tr>
<td>View-Toolbar</td>
<td>Show or hide the toolbar</td>
</tr>
<tr>
<td>View-Status Bar</td>
<td>Show or hide the status bar</td>
</tr>
<tr>
<td>View-Redraw Toolbar</td>
<td>Redraw the toolbar</td>
</tr>
<tr>
<td>Window-Cascade</td>
<td>Arrange windows so they overlap</td>
</tr>
<tr>
<td>Window-Tile Horizontally</td>
<td>Arrange windows as non-overlapping tiles</td>
</tr>
<tr>
<td>Window-Tile Vertically</td>
<td>Arrange windows as non-overlapping tiles</td>
</tr>
<tr>
<td>Window-Arrange Icons</td>
<td>Arrange icons at the bottom of the window</td>
</tr>
<tr>
<td>System-Setup Options</td>
<td>Open the system setup dialogue to edit advanced options</td>
</tr>
<tr>
<td>System-Update Common Words</td>
<td>Update the common words data object</td>
</tr>
<tr>
<td>System-Update Common Endings</td>
<td>Update the common endings data object</td>
</tr>
<tr>
<td>System-Destruct All Data Objects</td>
<td>Delete all data objects</td>
</tr>
<tr>
<td>System-Compare Extraction Methods</td>
<td>Compare the extraction results of the feature map and the table map method</td>
</tr>
<tr>
<td>Help-Index</td>
<td>List Help topics</td>
</tr>
<tr>
<td>Help-Context Help</td>
<td>Display help for clicked on buttons, menus and windows</td>
</tr>
<tr>
<td>Help-Using Help</td>
<td>Display instructions about how to use help</td>
</tr>
<tr>
<td>Help-System Info</td>
<td>Run the Microsoft Info application</td>
</tr>
<tr>
<td>Help-Run Demo</td>
<td>Select a demo to run</td>
</tr>
<tr>
<td>Help-About...</td>
<td>Display program information, version number and copyright</td>
</tr>
</tbody>
</table>
Evaluation [Appendices]

3.1.1 Data set composition

Query 1:  WAR*

Complete retrieval, presented in descending hit density (no fuzzy search was used):

<table>
<thead>
<tr>
<th>Reference</th>
<th>Document</th>
<th>Hits</th>
<th>Keywords</th>
</tr>
</thead>
<tbody>
<tr>
<td>W1</td>
<td>Q1.TXT</td>
<td>15</td>
<td>Warburg, investment bank, hostile takeover</td>
</tr>
<tr>
<td>W2</td>
<td>FZ.TXT</td>
<td>2</td>
<td>federation of taxpayers, warns, tax burden taxpayers</td>
</tr>
<tr>
<td>W3*</td>
<td>13V.TXT</td>
<td>7</td>
<td>warnings, Compaq, computer keyboards, wrist injuries</td>
</tr>
<tr>
<td>W4</td>
<td>G5.TXT</td>
<td>4</td>
<td>Time Warner inc, Viacom inc. sells theater chain</td>
</tr>
<tr>
<td>W5</td>
<td>1F6.TXT</td>
<td>2</td>
<td>fare war, eurotunnel denies fare reductions</td>
</tr>
<tr>
<td>W6</td>
<td>10L.TXT</td>
<td>23</td>
<td>marketing &amp; media, iced tea to europe, warner bros. records</td>
</tr>
<tr>
<td>W7</td>
<td>10G.TXT</td>
<td>6</td>
<td>Unilever, Omo power, detergent war</td>
</tr>
<tr>
<td>W8</td>
<td>AW.TXT</td>
<td>6</td>
<td>Cisco systems, communicating with small investors</td>
</tr>
<tr>
<td>W9</td>
<td>X8.TXT</td>
<td>4</td>
<td>pharmaceutical industry, takeover, Glaxo holdings</td>
</tr>
<tr>
<td>W10*</td>
<td>13W.TXT</td>
<td>2</td>
<td>IBM, order system for software</td>
</tr>
<tr>
<td>W11</td>
<td>1SK.TXT</td>
<td>1</td>
<td>Alliance Pharmaceutical, Johnson &amp; Johnson, drugs group</td>
</tr>
<tr>
<td>W12</td>
<td>10H.TXT</td>
<td>4</td>
<td>Unilever, Omo power, Proctor &amp; Gamble, soap war</td>
</tr>
<tr>
<td>W13</td>
<td>PX.TXT</td>
<td>3</td>
<td>Unilever pretax profit rose</td>
</tr>
<tr>
<td>W14</td>
<td>13O.TXT</td>
<td>1</td>
<td>wholesale prices fall in western Germany</td>
</tr>
<tr>
<td>W15</td>
<td>JM.TXT</td>
<td>4</td>
<td>car trip in Europe, reasons to stay at home</td>
</tr>
<tr>
<td>W16</td>
<td>AJ.TXT</td>
<td>6</td>
<td>business opportunity in Poland, financing</td>
</tr>
<tr>
<td>W17</td>
<td>AK.TXT</td>
<td>1</td>
<td>Russia plans to take action on plague of unpaid bills</td>
</tr>
<tr>
<td>W18</td>
<td>9T.TXT</td>
<td>5</td>
<td>Murdoch’s price war, newspaper sales up, profits are falling</td>
</tr>
<tr>
<td>W19</td>
<td>Q7.TXT</td>
<td>3</td>
<td>World Bank warned Turkey, international loans and credits</td>
</tr>
<tr>
<td>W20</td>
<td>Q2.TXT</td>
<td>2</td>
<td>Polish court, license, PolSat TV, nationwide broadcasting</td>
</tr>
<tr>
<td>W21</td>
<td>X3.TXT</td>
<td>2</td>
<td>U.S. industrial production +0.2% in July, warn weather</td>
</tr>
<tr>
<td>W22</td>
<td>70.TXT</td>
<td>1</td>
<td>Britain, Smart-card technology, drivers, EC directive</td>
</tr>
<tr>
<td>W23</td>
<td>9X.TXT</td>
<td>3</td>
<td>British monetary-policy, industrial production, Warburg</td>
</tr>
<tr>
<td>W24*</td>
<td>1HI.TXT</td>
<td>2</td>
<td>Hewlett-Packard share price rises on increases in earnings</td>
</tr>
<tr>
<td>W25</td>
<td>Q1.TXT</td>
<td>2</td>
<td>microwave filter, air warfare, communications applications</td>
</tr>
<tr>
<td>W26</td>
<td>10K.TXT</td>
<td>2</td>
<td>M6, French television, bourse listing</td>
</tr>
<tr>
<td>W27</td>
<td>FP.TXT</td>
<td>3</td>
<td>German teens find new fuel for disco raves, Red Bull</td>
</tr>
<tr>
<td>W28</td>
<td>FS.TXT</td>
<td>1</td>
<td>Kuwait seals Russian ties with major arms purchase</td>
</tr>
<tr>
<td>W29</td>
<td>AZ.TXT</td>
<td>2</td>
<td>China’s elusive effect on market for commodities</td>
</tr>
<tr>
<td>W30</td>
<td>AE.TXT</td>
<td>2</td>
<td>U.S. FCC, license, interactive television services</td>
</tr>
<tr>
<td>W31</td>
<td>D0.TXT</td>
<td>1</td>
<td>ENG Group, Bank Brussels Lambert, takeover</td>
</tr>
<tr>
<td>W32</td>
<td>CL.TXT</td>
<td>2</td>
<td>British Airways, rise in pretax profit, shares fall</td>
</tr>
<tr>
<td>W33</td>
<td>CK.TXT</td>
<td>1</td>
<td>Lending in U.K. to consumers rises to record</td>
</tr>
<tr>
<td>W34</td>
<td>AL.TXT</td>
<td>2</td>
<td>AIDS, epidemic, research, conference</td>
</tr>
<tr>
<td>W35</td>
<td>6K.TXT</td>
<td>2</td>
<td>CNN, BCC, Cox, 24-hour news business, competitors</td>
</tr>
<tr>
<td>W36</td>
<td>PM.TXT</td>
<td>2</td>
<td>NATO begins search for new top secretary-general</td>
</tr>
<tr>
<td>W37</td>
<td>PS.TXT</td>
<td>1</td>
<td>TBB, CAA, pact, video programming, telephone customers</td>
</tr>
<tr>
<td>W38</td>
<td>73.TXT</td>
<td>2</td>
<td>German postal service will offer 25% of shares to public</td>
</tr>
<tr>
<td>W39</td>
<td>9S.TXT</td>
<td>2</td>
<td>Europe’s economic recovery is beating expectations</td>
</tr>
<tr>
<td>W40</td>
<td>JN.TXT</td>
<td>2</td>
<td>Zurich, Fust Investors, takeover, insider trading</td>
</tr>
<tr>
<td>Reference</td>
<td>Document</td>
<td>Hits</td>
<td>Keywords</td>
</tr>
<tr>
<td>-----------</td>
<td>----------</td>
<td>------</td>
<td>----------</td>
</tr>
<tr>
<td>W41</td>
<td>71.TXT</td>
<td>1</td>
<td>costly diet products in Russia, Herbalife, scientific?</td>
</tr>
<tr>
<td>W42</td>
<td>CR.TXT</td>
<td>1</td>
<td>Saatchi &amp; Saatchi Co., profit, British advertising, marketing</td>
</tr>
<tr>
<td>W43</td>
<td>1HD.TXT</td>
<td>3</td>
<td>American Express, data mining, Sybase Inc</td>
</tr>
<tr>
<td>W44*</td>
<td>A0.TXT</td>
<td>1</td>
<td>IBM’s overhaul of disk-drive unit may cut jobs in Europe</td>
</tr>
<tr>
<td>W45</td>
<td>AQ.TXT</td>
<td>1</td>
<td>Union Bank of Switzerland, shares drop 28%</td>
</tr>
<tr>
<td>W46</td>
<td>A1.TXT</td>
<td>1</td>
<td>Finalists to purchase Kodak’s household-products division</td>
</tr>
<tr>
<td>W47</td>
<td>CX.TXT</td>
<td>3</td>
<td>China should take a hard line on software pirates</td>
</tr>
<tr>
<td>W48</td>
<td>CH.TXT</td>
<td>1</td>
<td>Italian Silvio Berlusconi, television advertising, RAI</td>
</tr>
<tr>
<td>W49</td>
<td>GD.TXT</td>
<td>1</td>
<td>German teens find new fuel for discos, continued</td>
</tr>
<tr>
<td>W50</td>
<td>XL.TXT</td>
<td>2</td>
<td>Toys R Us, investors, Petrie Stores, deal, Warburg complex</td>
</tr>
<tr>
<td>W51</td>
<td>146.TXT</td>
<td>2</td>
<td>new markets in central and eastern europe, investors</td>
</tr>
<tr>
<td>W52</td>
<td>CQ.TXT</td>
<td>1</td>
<td>Germany, Bayer AG, buying drug business of Kodak</td>
</tr>
<tr>
<td>W53</td>
<td>6Y.TXT</td>
<td>1</td>
<td>global news business, BBC challenges CNN</td>
</tr>
<tr>
<td>W54</td>
<td>AR.TXT</td>
<td>1</td>
<td>European central banks, Germany’s Bundesbank, discount</td>
</tr>
<tr>
<td>W55</td>
<td>1JN.TXT</td>
<td>2</td>
<td>Russian mafia, new problems, La Cosa Nostra</td>
</tr>
<tr>
<td>W56</td>
<td>1JP.TXT</td>
<td>1</td>
<td>American Home Products buys American Cyanamid</td>
</tr>
<tr>
<td>W57</td>
<td>10B.TXT</td>
<td>1</td>
<td>German drug investigation, two Schering AG drugs</td>
</tr>
<tr>
<td>W58</td>
<td>X2.TXT</td>
<td>1</td>
<td>Portuguese Bank at war, free-market policies</td>
</tr>
<tr>
<td>W59</td>
<td>QA.TXT</td>
<td>1</td>
<td>Algeria, nationalist guerrillas, Islamic slogans, war</td>
</tr>
<tr>
<td>W60</td>
<td>WZ.TXT</td>
<td>1</td>
<td>Carlos the Jackal, arrest, terrorism, politics, Sudan, France</td>
</tr>
<tr>
<td>W61</td>
<td>FQ.TXT</td>
<td>1</td>
<td>French cognac, consumption dropped, Norwegians buy cars</td>
</tr>
<tr>
<td>W62</td>
<td>X1.TXT</td>
<td>1</td>
<td>trends, business travel and tourism, economic recovery</td>
</tr>
<tr>
<td>W63</td>
<td>PL.TXT</td>
<td>1</td>
<td>Maastricht, fiscal policy, Europe, monetary union</td>
</tr>
<tr>
<td>W64</td>
<td>6H.TXT</td>
<td>1</td>
<td>scandals, Washington, Congress, lobbyist-paid trips</td>
</tr>
<tr>
<td>W65</td>
<td>1JU.TXT</td>
<td>1</td>
<td>LDDS Communications inc., telephone, acquisition, WiTcl</td>
</tr>
<tr>
<td>W66</td>
<td>10A.TXT</td>
<td>1</td>
<td>Russia, Germany, plutonium smuggling, pressure</td>
</tr>
<tr>
<td>W67*</td>
<td>CG.TXT</td>
<td>2</td>
<td>Reebok, workers’ rights, China, software, aquarium, fish</td>
</tr>
<tr>
<td>W69**</td>
<td>A3.TXT</td>
<td>1</td>
<td>Dell Computers, return, notebook, new designs, price battle</td>
</tr>
<tr>
<td>W70</td>
<td>13S.TXT</td>
<td>1</td>
<td>Scandinavian Airlines System, recovery, pretax profit</td>
</tr>
<tr>
<td>W71</td>
<td>Q4.TXT</td>
<td>1</td>
<td>record stores, changing, multimedia stores</td>
</tr>
<tr>
<td>W72</td>
<td>G7.TXT</td>
<td>1</td>
<td>cars, Renault SA, front-runner in French privatization race</td>
</tr>
<tr>
<td>W73</td>
<td>HG.TXT</td>
<td>1</td>
<td>U.S. retailing, Dayton Hudson, Wal-Mart Stores, earnings</td>
</tr>
<tr>
<td>W74</td>
<td>JO.TXT</td>
<td>1</td>
<td>Johnson &amp; Johnson acquires Neutrogena, personal care</td>
</tr>
<tr>
<td>W75</td>
<td>HG.TXT</td>
<td>1</td>
<td>U.S. health care, problem, subsidized by federal government</td>
</tr>
<tr>
<td>W76</td>
<td>13G.TXT</td>
<td>1</td>
<td>Vietnam places hope for economic health in local enterprise</td>
</tr>
<tr>
<td>W77</td>
<td>60.TXT</td>
<td>1</td>
<td>American Cyanamid, American Home Products, takeover</td>
</tr>
<tr>
<td>W78</td>
<td>JK.TXT</td>
<td>1</td>
<td>Japanese price revolution, consumer behaviour, shopping</td>
</tr>
<tr>
<td>W79</td>
<td>13U.TXT</td>
<td>1</td>
<td>technology &amp; health, employees, virtual offices, low morale</td>
</tr>
<tr>
<td>W80</td>
<td>GA.TXT</td>
<td>2</td>
<td>Boston sees payoff and problems in east Europe, expanding</td>
</tr>
<tr>
<td>W81</td>
<td>B1.TXT</td>
<td>1</td>
<td>Asian markets, bourses consolidate after gains, Tokyo</td>
</tr>
<tr>
<td>W82</td>
<td>PT.TXT</td>
<td>1</td>
<td>Bill Clinton, defeat on Crime Bill, Washington, Congress</td>
</tr>
<tr>
<td>W83</td>
<td>1HU.TXT</td>
<td>1</td>
<td>Helsinki, Oy Nokia, toilet-paper, cellular phones</td>
</tr>
<tr>
<td>W84</td>
<td>Q3.TXT</td>
<td>1</td>
<td>British, high taxes, bargain hunters dent U.K. alcohol sales</td>
</tr>
<tr>
<td>W85</td>
<td>B0.TXT</td>
<td>1</td>
<td>Eurobond Market, quiet week</td>
</tr>
<tr>
<td>W86</td>
<td>GK.TXT</td>
<td>2</td>
<td>international bond indexes, bund prices fall</td>
</tr>
</tbody>
</table>
**Query 2:**  
EC  
Complete retrieval, presented in descending hit density (no fuzzy search was used):

<table>
<thead>
<tr>
<th>Reference</th>
<th>Document</th>
<th>Hits</th>
<th>Keywords</th>
</tr>
</thead>
<tbody>
<tr>
<td>E1</td>
<td>13J.TXT</td>
<td>1</td>
<td>EC lobbyists, American Express, Brussels</td>
</tr>
<tr>
<td>E2</td>
<td>FY.TXT</td>
<td>1</td>
<td>Terra Industries buys fertilizer products concern (NO ec!)</td>
</tr>
<tr>
<td>E3</td>
<td>10I.TXT</td>
<td>1</td>
<td>marketing &amp; media, iced tea to europe, warner bros. records</td>
</tr>
</tbody>
</table>

**Query 3:**  
COMPUT*  
Incomplete retrieval (the 18 highest ranked documents out of 50 are selected), presented in descending hit density (no fuzzy search was used):

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

Documents with multiple occurrences

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

**Correlation assignment**

** 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.
3.1.2.1 Full-text query

C1 (1F7.TXT)

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-performance 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 carrying over into the corporate side," said 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-
advanced power management, enhanced networking capabilities and easier-to-use “plug In’ play” features.

3.1.2.2 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”

3.1.3 Settings composition

Used configurations in the preliminary evaluation process

<table>
<thead>
<tr>
<th>Generalisation factor</th>
<th>Context size</th>
<th>Space is a character</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>3</td>
<td>no</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>no</td>
</tr>
<tr>
<td>6</td>
<td>3</td>
<td>no</td>
</tr>
<tr>
<td>2</td>
<td>5</td>
<td>no</td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td>no</td>
</tr>
<tr>
<td>6</td>
<td>5</td>
<td>no</td>
</tr>
<tr>
<td>2</td>
<td>7</td>
<td>no</td>
</tr>
<tr>
<td>4</td>
<td>7</td>
<td>no</td>
</tr>
<tr>
<td>6</td>
<td>7</td>
<td>no</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Generalisation factor</th>
<th>Context size</th>
<th>Space is a character</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>3</td>
<td>yes</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>yes</td>
</tr>
<tr>
<td>6</td>
<td>3</td>
<td>yes</td>
</tr>
<tr>
<td>2</td>
<td>5</td>
<td>yes</td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td>yes</td>
</tr>
<tr>
<td>6</td>
<td>5</td>
<td>yes</td>
</tr>
<tr>
<td>2</td>
<td>7</td>
<td>yes</td>
</tr>
<tr>
<td>4</td>
<td>7</td>
<td>yes</td>
</tr>
<tr>
<td>6</td>
<td>7</td>
<td>yes</td>
</tr>
</tbody>
</table>

N.B. In the case of the artificial query all settings with Space is a character = yes have not been processed. This parameter can be used to include a word’s natural context in the representation process. In an artificial query there is by definition no natural context.
Example of a settings configuration (FTQ23106.SET)

PARAMETERS:

Context size : 3
Sensors per neuron : 3
X-dimension of feature map : 13
Y-dimension of feature map : 17
Initial weight spread value : 15
Number of objects : 27
Vector spread value : 2
Maximum error recording capacity : 8220
Error compression : 1
Learn cycles : 8220
Sigma start (start region size) : 4.500000
Sigma end (end region size) : 0.500000
Epsilon start (start learning rate) : 0.950000
Epsilon end (end learning rate) : 0.050000
Perfect hit threshold : 0.061100
Recordset view threshold : 0.400000
Text part separator :
Paint interval : 500
Line unit size : 30
Line accentuation in visualisation of interneuronal distances : 100.000000
Visualisation self-organisation from dimension : 0
Visualisation self-organisation to dimension : 1
Minimum word length : 3
[DEMO] Shuffle vector set before training process : no
[DEMO] Number of shuffles : 0
[DEMO] Include extraction process : no
[DEMO] Print feature map before training process : no
[DEMO] Print vector map before training process : no
[DEMO] Print error recording after training process : no
[DEMO] Print activity recording after extraction process : no

PREFERENCES:

Activate system menu : yes
Activate files and paths menu : yes
Use decreasing epsilon in learn process : yes
Use separator in extraction process : no
Use table map in extraction process : yes
Divide words with one space : yes
Filter small words : yes
Filter common words : yes
Filter common endings : yes
Print pattern errors : no
Print text part errors : no
Print current data flow : no
Tile picture vertically at setup : no
Visualise self-organisation : no
Visualise interneuronal distances: no
Visualise object distribution: no
Visualise error recording: no
Visualise activity recording: no

FILES AND PATHS:
Query file: C:\ETC\FILTER\DATASET\1F7.TXT
Query file activation: yes
Data files:
C:\ETC\FILTER\DATASET\1F9.TXT
C:\ETC\FILTER\DATASET\1QV.TXT
C:\ETC\FILTER\DATASET\XJ.TXT
[...another 97 documents...]
Data files activation: yes
Add-Dir extension: *.TXT
Common words file: C:\MRS\MSVC\FILTER\TXT\OCR_COMM.TXT
Common words file activation: yes
Common endings file: C:\MRS\MSVC\FILTER\TXT\ENDING.TXT
Common endings file activation: yes
Output directory: C:\TMP
Application directory: C:\MRS\MSVC\FILTER\SET\FTQ23106.SET

SYSTEM:
Password: **********
Execution pause: 6
Hide execution: yes
Use password mechanism: no
Settings Files Evaluation:

3.2 Expectations

Input filter: common words
Besides the usual common words, some special words have been added too. These words were added to the original text by the scanning software. These extra noise words are marked here with an asterisk.
him  himself  how  l  if  image*  in
index*  into  is  it  its  j  journal*
jjust  k  l  like  likes  m  made
make  makes  many  me  might  more  most
much  must  my  n  never  not  now
of  on  only  or  other  others
our  ours  out  over  p  q  r
s  said  same  scan*  see  sees  she
should  since  so  some  source*  still  street*
such  t  take  takes  than  that  the
their  them  then  there  these  they  this
title*  those  through  tif*  tiff*  to  too
u  under  up  v  very  w  wall*
was  way  we  well  were  what  when
where  which  while  who  with  would  x
xref*  y  you  your  yours  z

Input filter: common endings
ably  ibly  ily  ss  ous  ies  s
ted  ed  ing  e  al  ion  ary
ability  ibility  ility  ify  abl  ibl  iv
at  is  ific  olv

Full-text query’s internal data flow
(The newline symbol has been added manually for clarification:)

\nEUROP\n\nDELLPLANOVERHAUL
DESKTOPCOMPUTER
AIMCOMPAN
\nSTAFFREPORTER
AUSTINTXADENCOMPUTERCORP
EXPECTANNOUNCTODAYOVERHAUL
HIGHENDOPTIPLEXLINCORPORAT
DESKTOPCOMPUTERWILLINCLUDPRIC
CUTINTRODUCTHIGHPERFORM
ANCPECTIUMMICROPROCESSOR
COMPUTERVENDOR
BREAKPRICBARRIEROFFER
FULLYCONFIGUREDTSKOPSYST
BASSPEEDYMEGAHERTZVERS
INTELCORPPENTIUMCHIP
MOVETOWARDPENTIUMCARRY
CORPORATSID
DOUGMACGREGORDELLVICPRESIDENT
DESKTOPCOMPUTERNONCORPORAT
CUSTOMERQUESTWHETHER
PENTIUMQUEST

DELLPUSHHARDNEWEST
INTELCHEWPHALF
DIMENSIONSMACHINESOLDHOMUSER
SMALLBUSINESSEPENTIUMCHIP
MACGREGOR
CORPORATCUSTOMER
WAITPENTIUMPRICEFALL
NEWOPTIPLEXPRICEBUSINESS
WILLABLBYPENTIUMBASMACHINE
PRICESIMILAR
PAYYEARAGOSLOWER
BASCOMPUTERMACGREGOR

DELLNEWOPTIPLEXMODEL
REPLACEMACHINEINTRODUCCYEAR
AGONEWMACHINEINCORPORAT
VANCPOWERMANAGEMENTENHANCNET
WORKCAPABILITEASIERPLUG
PLAYFEATURE

Artificial query's internal data flow
DELLDESKTOPCOMPUTEROPTIPLEXLIN
DESKTOPCOMPUTERT INTRODUCETHIGHPERFORMANC
PENTIUMMICROPROCESSORCOMPUTERPRICBARRIER
CONFIGURDESKTOPSYSTEMSPEMEGAHERTZ
INTELPENTIUMCHIPNEWDOUGMACGREGORDELL
DESKTOPCOMPUTERPENTIUMDELLINTELCHIP
PENTIUMCHIPMACGREGORPENTIUMOPTIPLEX
PENTIUMBASMACHINEBASCOMPUTER
OPTIPLEXMODELPOWERMANAGEMENTNETWORK
PLUGPLAY
3.3.1 Preliminary Results

Measurement
There are 14 documents that may be retrieved to continue the state of maximum precision: C1, C2, C5, C8, C3, C4, C6, C7, C11, C14, C15, C16, C17 and W67. The first 4 documents must be retrieved to reach the state of maximum recall. One of these four documents is the query document itself.

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 = 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
Pr: Maximum number of retrieved documents at 100% precision with hitlist sorted on x
Rx: Minimum number of retrieved documents at 100% recall with hitlist sorted on x
Ex: Minimum number of irrelevant documents in hitlist at 100% recall with hitlist sorted on x
x1: Number of documents with hitlist sorted on Average error
x2: Number of documents with hitlist sorted on Perfect hit rate
x3: Number of documents with hitlist sorted on Average hit error

<table>
<thead>
<tr>
<th>Settings</th>
<th>Q. error</th>
<th>D. error</th>
<th>P1</th>
<th>R1</th>
<th>E1</th>
<th>P2</th>
<th>R2</th>
<th>E2</th>
<th>P3</th>
<th>R3</th>
<th>E3</th>
</tr>
</thead>
<tbody>
<tr>
<td>FTQ23020</td>
<td>0.119695</td>
<td>0.177454</td>
<td>1</td>
<td>11</td>
<td>2</td>
<td>2</td>
<td>13</td>
<td>6</td>
<td>1</td>
<td>80</td>
<td>70</td>
</tr>
<tr>
<td>FTQ43020</td>
<td>0.159510</td>
<td>0.206315</td>
<td>2</td>
<td>14</td>
<td>6</td>
<td>4</td>
<td>18</td>
<td>9</td>
<td>1</td>
<td>37</td>
<td>26</td>
</tr>
<tr>
<td>FTQ63020</td>
<td>0.181064</td>
<td>0.217626</td>
<td>1</td>
<td>11</td>
<td>2</td>
<td>1</td>
<td>11</td>
<td>4</td>
<td>1</td>
<td>18</td>
<td>11</td>
</tr>
<tr>
<td>FTQ25020</td>
<td>0.179135</td>
<td>0.259371</td>
<td>1</td>
<td>13</td>
<td>3</td>
<td>5</td>
<td>10</td>
<td>2</td>
<td>3</td>
<td>9</td>
<td>3</td>
</tr>
<tr>
<td>FTQ45020</td>
<td>0.215698</td>
<td>0.275734</td>
<td>2</td>
<td>14</td>
<td>4</td>
<td>4</td>
<td>13</td>
<td>5</td>
<td>2</td>
<td>15</td>
<td>6</td>
</tr>
<tr>
<td>FTQ65020</td>
<td>0.234187</td>
<td>0.287028</td>
<td>1</td>
<td>14</td>
<td>4</td>
<td>4</td>
<td>8</td>
<td>1</td>
<td>1</td>
<td>10</td>
<td>3</td>
</tr>
<tr>
<td>FTQ27020</td>
<td>0.219438</td>
<td>0.301106</td>
<td>1</td>
<td>11</td>
<td>3</td>
<td>6</td>
<td>6</td>
<td>0</td>
<td>5</td>
<td>5</td>
<td>0</td>
</tr>
<tr>
<td>FTQ47020</td>
<td>0.249470</td>
<td>0.310746</td>
<td>1</td>
<td>15</td>
<td>5</td>
<td>7</td>
<td>7</td>
<td>0</td>
<td>7</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>FTQ67020</td>
<td>0.268693</td>
<td>0.318401</td>
<td>1</td>
<td>14</td>
<td>5</td>
<td>6</td>
<td>6</td>
<td>0</td>
<td>6</td>
<td>6</td>
<td>0</td>
</tr>
<tr>
<td>FTQ23120</td>
<td>0.120582</td>
<td>0.168126</td>
<td>6</td>
<td>6</td>
<td>0</td>
<td>2</td>
<td>14</td>
<td>5</td>
<td>1</td>
<td>70</td>
<td>59</td>
</tr>
<tr>
<td>FTQ43120</td>
<td>0.155662</td>
<td>0.200007</td>
<td>3</td>
<td>8</td>
<td>1</td>
<td>1</td>
<td>15</td>
<td>8</td>
<td>1</td>
<td>34</td>
<td>25</td>
</tr>
<tr>
<td>FTQ63120</td>
<td>0.182588</td>
<td>0.217062</td>
<td>2</td>
<td>11</td>
<td>2</td>
<td>1</td>
<td>21</td>
<td>12</td>
<td>1</td>
<td>34</td>
<td>25</td>
</tr>
<tr>
<td>FTQ25120</td>
<td>0.183954</td>
<td>0.252867</td>
<td>3</td>
<td>7</td>
<td>1</td>
<td>6</td>
<td>8</td>
<td>1</td>
<td>3</td>
<td>10</td>
<td>4</td>
</tr>
<tr>
<td>FTQ45120</td>
<td>0.218768</td>
<td>0.253276</td>
<td>4</td>
<td>8</td>
<td>2</td>
<td>7</td>
<td>7</td>
<td>0</td>
<td>2</td>
<td>6</td>
<td>1</td>
</tr>
<tr>
<td>FTQ65120</td>
<td>0.240167</td>
<td>0.286132</td>
<td>3</td>
<td>7</td>
<td>1</td>
<td>4</td>
<td>9</td>
<td>1</td>
<td>2</td>
<td>8</td>
<td>1</td>
</tr>
<tr>
<td>FTQ27120</td>
<td>0.228325</td>
<td>0.300309</td>
<td>4</td>
<td>8</td>
<td>2</td>
<td>8</td>
<td>8</td>
<td>0</td>
<td>5</td>
<td>8</td>
<td>1</td>
</tr>
<tr>
<td>FTQ47120</td>
<td>0.257845</td>
<td>0.312278</td>
<td>4</td>
<td>8</td>
<td>2</td>
<td>6</td>
<td>6</td>
<td>0</td>
<td>6</td>
<td>6</td>
<td>0</td>
</tr>
<tr>
<td>FTQ67120</td>
<td>0.276522</td>
<td>0.321395</td>
<td>4</td>
<td>10</td>
<td>3</td>
<td>6</td>
<td>6</td>
<td>0</td>
<td>6</td>
<td>6</td>
<td>0</td>
</tr>
</tbody>
</table>
### Settings

<table>
<thead>
<tr>
<th>Settings</th>
<th>Q. error</th>
<th>D. error</th>
<th>P1</th>
<th>R1</th>
<th>E1</th>
<th>P2</th>
<th>R2</th>
<th>E2</th>
<th>P3</th>
<th>R3</th>
<th>E3</th>
</tr>
</thead>
<tbody>
<tr>
<td>AQ23020</td>
<td>0.202122</td>
<td>0.226122</td>
<td>4</td>
<td>15</td>
<td>5</td>
<td>2</td>
<td>28</td>
<td>16</td>
<td>2</td>
<td>58</td>
<td>45</td>
</tr>
<tr>
<td>AQ43020</td>
<td>0.232747</td>
<td>0.249897</td>
<td>3</td>
<td>18</td>
<td>9</td>
<td>4</td>
<td>28</td>
<td>16</td>
<td>4</td>
<td>56</td>
<td>43</td>
</tr>
<tr>
<td>AQ63020</td>
<td>0.253562</td>
<td>0.267204</td>
<td>2</td>
<td>30</td>
<td>18</td>
<td>2</td>
<td>20</td>
<td>9</td>
<td>1</td>
<td>30</td>
<td>18</td>
</tr>
<tr>
<td>AQ25020</td>
<td>0.268683</td>
<td>0.295165</td>
<td>3</td>
<td>12</td>
<td>3</td>
<td>7</td>
<td>7</td>
<td>0</td>
<td>6</td>
<td>8</td>
<td>1</td>
</tr>
<tr>
<td>AQ45020</td>
<td>0.284676</td>
<td>0.308447</td>
<td>3</td>
<td>14</td>
<td>5</td>
<td>6</td>
<td>6</td>
<td>0</td>
<td>4</td>
<td>8</td>
<td>2</td>
</tr>
<tr>
<td>AQ65020</td>
<td>0.299796</td>
<td>0.320611</td>
<td>2</td>
<td>25</td>
<td>15</td>
<td>8</td>
<td>8</td>
<td>0</td>
<td>5</td>
<td>8</td>
<td>1</td>
</tr>
<tr>
<td>AQ27020</td>
<td>0.315326</td>
<td>0.337489</td>
<td>1</td>
<td>30</td>
<td>19</td>
<td>8</td>
<td>10</td>
<td>1</td>
<td>7</td>
<td>11</td>
<td>2</td>
</tr>
<tr>
<td>AQ47020</td>
<td>0.328301</td>
<td>0.344352</td>
<td>2</td>
<td>37</td>
<td>25</td>
<td>6</td>
<td>13</td>
<td>3</td>
<td>5</td>
<td>16</td>
<td>6</td>
</tr>
</tbody>
</table>

---

**Precision and recall, sorted on average error**

This chart contains all preliminary outcomes of the full-text query without space as a character, sorted on the average error:
**Precision and recall, sorted on perfect hit rate**

This chart contains all preliminary outcomes of the full-text query without space as a character, sorted on the perfect hit rate:

![Precision Recall Chart](chart1.png)

**Precision and recall, sorted on average hit error**

This chart contains all preliminary outcomes of the full-text query without space as a character, sorted on the average hit error:

![Precision Recall Chart](chart2.png)
**Precision and recall, sorted on average error**

This chart contains all preliminary outcomes of the full-text query with space as a character, sorted on the average error:

![Graph showing precision and recall on average error](chart1.png)

**Precision and recall, sorted on perfect hit rate**

This chart contains all preliminary outcomes of the full-text query with space as a character, sorted on the perfect hit rate:

![Graph showing precision and recall on perfect hit rate](chart2.png)
Precision and recall, sorted on average hit error
This chart contains all preliminary outcomes of the full-text query with space as a character, sorted on the average hit error:

Precision and recall, sorted on average error
This chart contains all preliminary outcomes of the artificial query, sorted on the average error:
Precision and recall, sorted on perfect hit rate

This chart contains all preliminary outcomes of the artificial query, sorted on the perfect hit rate:

![Precision and recall, sorted on perfect hit rate](image)

Precision and recall, sorted on average hit error

This chart contains all preliminary outcomes of the artificial query, sorted on the average hit error:

![Precision and recall, sorted on average hit error](image)
3.3.3 Additional Results

(For completeness the introduction of Appendix 3.3.1 is repeated here.)

Measurement

There are 14 documents that may be retrieved to continue the state of maximum precision: C1, C2, C5, C8, C3, C4, C6, C7, C11, C14, C15, C16, C17 and W67. The first 4 documents must be retrieved to reach the state of maximum recall. One of these four documents is the query document itself.

Table abbreviations

- **FTQ** = Full-text query
- **AQ** = Artificial query
- **Digit 1** = Generalisation factor
- **Digit 2** = Context size
- **Digit 3** = Space as character
- **Digit 4,5** = 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
- **P:** Maximum number of retrieved documents at 100% precision with hitlist sorted on $x$
- **R:** Minimum number of retrieved documents at 100% recall with hitlist sorted on $x$
- **E:** Minimum number of irrelevant documents in hitlist at 100% recall with hitlist sorted on $x$

x1: Number of documents with hitlist sorted on Average error

x2: Number of documents with hitlist sorted on Perfect hit rate

x3: Number of documents with hitlist sorted on Average hit error

<table>
<thead>
<tr>
<th>Settings</th>
<th>Q. error</th>
<th>D. error</th>
<th>P1</th>
<th>R1</th>
<th>E1</th>
<th>P2</th>
<th>R2</th>
<th>E2</th>
<th>P3</th>
<th>R3</th>
<th>E3</th>
</tr>
</thead>
<tbody>
<tr>
<td>FTQ27030</td>
<td>0.219438</td>
<td>0.301106</td>
<td>1</td>
<td>11</td>
<td>3</td>
<td>1</td>
<td>16</td>
<td>9</td>
<td>1</td>
<td>20</td>
<td>12</td>
</tr>
<tr>
<td>FTQ27010</td>
<td>0.219438</td>
<td>0.301106</td>
<td>1</td>
<td>11</td>
<td>3</td>
<td>6</td>
<td>6</td>
<td>0</td>
<td>6</td>
<td>6</td>
<td>0</td>
</tr>
<tr>
<td>FTQ23012</td>
<td>0.122862</td>
<td>0.180017</td>
<td>1</td>
<td>11</td>
<td>2</td>
<td>2</td>
<td>7</td>
<td>1</td>
<td>1</td>
<td>10</td>
<td>4</td>
</tr>
<tr>
<td>FTQ23009</td>
<td>0.122862</td>
<td>0.180017</td>
<td>1</td>
<td>11</td>
<td>2</td>
<td>9</td>
<td>1</td>
<td>1</td>
<td>10</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>FTQ23006</td>
<td>0.122862</td>
<td>0.180017</td>
<td>1</td>
<td>11</td>
<td>2</td>
<td>6</td>
<td>6</td>
<td>0</td>
<td>7</td>
<td>7</td>
<td>0</td>
</tr>
<tr>
<td>FTQ23109</td>
<td>0.120582</td>
<td>0.168126</td>
<td>6</td>
<td>6</td>
<td>0</td>
<td>6</td>
<td>6</td>
<td>0</td>
<td>4</td>
<td>7</td>
<td>1</td>
</tr>
<tr>
<td>FTQ23106</td>
<td>0.120582</td>
<td>0.168126</td>
<td>6</td>
<td>6</td>
<td>0</td>
<td>4</td>
<td>4</td>
<td>0</td>
<td>4</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>FTQ43107</td>
<td>0.155662</td>
<td>0.200007</td>
<td>3</td>
<td>8</td>
<td>1</td>
<td>23</td>
<td>11</td>
<td>1</td>
<td>36</td>
<td>24</td>
<td></td>
</tr>
<tr>
<td>FTQ43111</td>
<td>0.155662</td>
<td>0.200007</td>
<td>3</td>
<td>8</td>
<td>1</td>
<td>6</td>
<td>6</td>
<td>0</td>
<td>5</td>
<td>9</td>
<td>2</td>
</tr>
<tr>
<td>AQ130200</td>
<td>0.176395</td>
<td>0.205944</td>
<td>5</td>
<td>10</td>
<td>2</td>
<td>4</td>
<td>19</td>
<td>9</td>
<td>1</td>
<td>47</td>
<td>35</td>
</tr>
<tr>
<td>AQ130176</td>
<td>0.176395</td>
<td>0.205944</td>
<td>5</td>
<td>10</td>
<td>2</td>
<td>5</td>
<td>15</td>
<td>5</td>
<td>1</td>
<td>39</td>
<td>29</td>
</tr>
<tr>
<td>AQ130132</td>
<td>0.176395</td>
<td>0.205944</td>
<td>5</td>
<td>10</td>
<td>2</td>
<td>9</td>
<td>10</td>
<td>1</td>
<td>2</td>
<td>28</td>
<td>18</td>
</tr>
<tr>
<td>AQ130088</td>
<td>0.176395</td>
<td>0.205944</td>
<td>5</td>
<td>10</td>
<td>2</td>
<td>8</td>
<td>8</td>
<td>0</td>
<td>7</td>
<td>13</td>
<td>5</td>
</tr>
<tr>
<td>AQ130044</td>
<td>0.176395</td>
<td>0.205944</td>
<td>5</td>
<td>10</td>
<td>2</td>
<td>8</td>
<td>8</td>
<td>0</td>
<td>7</td>
<td>7</td>
<td>0</td>
</tr>
<tr>
<td>AQ130022</td>
<td>0.176395</td>
<td>0.205944</td>
<td>5</td>
<td>10</td>
<td>2</td>
<td>8</td>
<td>8</td>
<td>0</td>
<td>8</td>
<td>8</td>
<td>0</td>
</tr>
<tr>
<td>AQ150200</td>
<td>0.237150</td>
<td>0.279872</td>
<td>3</td>
<td>13</td>
<td>3</td>
<td>7</td>
<td>11</td>
<td>2</td>
<td>3</td>
<td>13</td>
<td>4</td>
</tr>
</tbody>
</table>
Precision and recall, sorted on perfect hit rate

This chart contains all additional outcomes of the full-text query, sorted on the perfect hit rate:
Precision and recall, sorted on average hit error
This chart contains all additional outcomes of the full-text query, sorted on the average hit error:

![Graph showing precision and recall on average hit error]

Precision and recall, sorted on perfect hit rate
This chart contains all additional outcomes of the artificial query, sorted on the perfect hit rate:

![Graph showing precision and recall on perfect hit rate]
Precision and recall, sorted on average hit error

This chart contains all additional outcomes of the artificial query, sorted on the average hit error:

3.4 Comparison

(For completeness the introduction of Appendix 3.3.1 is repeated here again.)

Measurement

There are 14 documents that may be retrieved to continue the state of maximum precision: C1, C2, C5, C8, C3, C4, C6, C7, C11, C14, C15, C16, C17 and W67. The first 4 documents must be retrieved to reach the state of maximum recall. One of these four documents is the query document itself.

All hitlists have been sorted on hit density before examination because this is the most sophisticated relevance ranking method available in ZyIMAGE.

Table abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Query</td>
<td>ZyIMAGE query</td>
</tr>
<tr>
<td>F</td>
<td>Fuzzy degree used</td>
</tr>
<tr>
<td>P</td>
<td>Maximum number of retrieved documents at 100% precision</td>
</tr>
<tr>
<td>R</td>
<td>Minimum number of retrieved documents at 100% recall</td>
</tr>
<tr>
<td>E</td>
<td>Minimum number of irrelevant documents at 100% recall</td>
</tr>
</tbody>
</table>
Comment: Notable observations
{ ... } : Identical contents to the cell in the previous row, same column

Table of results (ZyIMAGE Boolean queries)

<table>
<thead>
<tr>
<th>Query</th>
<th>F</th>
<th>P</th>
<th>R</th>
<th>E</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>COMPUT*</td>
<td>0-4</td>
<td>8</td>
<td>8</td>
<td>0</td>
<td>data composition query</td>
</tr>
<tr>
<td>COMPUT* OR DELL</td>
<td>0</td>
<td>6</td>
<td>6</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>{ ... }</td>
<td>1</td>
<td>6</td>
<td>10</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>{ ... }</td>
<td>2-4</td>
<td>2</td>
<td>9</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>COMPUT* OR DELL*</td>
<td>0</td>
<td>6</td>
<td>6</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>(COMPUT* OR DELL) AND (NOT KEYB*)</td>
<td>0-4</td>
<td>6</td>
<td>-</td>
<td>-</td>
<td>1 essential document was not found</td>
</tr>
<tr>
<td>{ ... }</td>
<td>1</td>
<td>4</td>
<td>-</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>{ ... }</td>
<td>2-4</td>
<td>1</td>
<td>-</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>(COMPUT* OR DELL) AND (NOT SOFTW*)</td>
<td>0</td>
<td>5</td>
<td>5</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>{ ... }</td>
<td>1</td>
<td>4</td>
<td>7</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>{ ... }</td>
<td>2-4</td>
<td>2</td>
<td>8</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>(COMPUT* OR DELL*) AND (LINE OR DESKTOP)</td>
<td>0</td>
<td>4</td>
<td>4</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>{ ... }</td>
<td>1-4</td>
<td>5</td>
<td>5</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>(COMPUT* OR DELL) AND (LINE OR DESKTOP)</td>
<td>0</td>
<td>4</td>
<td>4</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>{ ... }</td>
<td>1</td>
<td>5</td>
<td>5</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>{ ... }</td>
<td>2-4</td>
<td>3</td>
<td>5</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>COMPUT* AND (LINE OR DESKTOP OR DELL*)</td>
<td>0</td>
<td>4</td>
<td>4</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>{ ... }</td>
<td>1-4</td>
<td>5</td>
<td>5</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>COMPUT* AND (LINE OR DESKTOP OR DELL)</td>
<td>0</td>
<td>4</td>
<td>4</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>{ ... }</td>
<td>1</td>
<td>5</td>
<td>5</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>{ ... }</td>
<td>2-4</td>
<td>4</td>
<td>4</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

Table of results (ZyIMAGE quorum queries)

<table>
<thead>
<tr>
<th>Query</th>
<th>F</th>
<th>P</th>
<th>R</th>
<th>E</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 of {Dell, Desktop, Computers, OptiFlex, 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}</td>
<td>0</td>
<td>5</td>
<td>5</td>
<td>0</td>
</tr>
<tr>
<td>{ ... }</td>
<td>1</td>
<td>1</td>
<td>5</td>
<td>1</td>
</tr>
<tr>
<td>{ ... }</td>
<td>2, 4</td>
<td>1</td>
<td>6</td>
<td>1</td>
</tr>
<tr>
<td>{ ... }</td>
<td>3</td>
<td>1</td>
<td>7</td>
<td>1</td>
</tr>
<tr>
<td>2 of {Dell, Desktop, Computers, OptiFlex, 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}</td>
<td>0</td>
<td>5</td>
<td>5</td>
<td>0</td>
</tr>
<tr>
<td>{ ... }</td>
<td>1</td>
<td>1</td>
<td>5</td>
<td>1</td>
</tr>
<tr>
<td>{ ... }</td>
<td>2, 4</td>
<td>1</td>
<td>6</td>
<td>1</td>
</tr>
<tr>
<td>{ ... }</td>
<td>3</td>
<td>1</td>
<td>7</td>
<td>1</td>
</tr>
<tr>
<td>3 of {Dell, Desktop, Computers, OptiFlex, 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}</td>
<td>0</td>
<td>5</td>
<td>5</td>
<td>0</td>
</tr>
<tr>
<td>{ ... }</td>
<td>1</td>
<td>6</td>
<td>6</td>
<td>0</td>
</tr>
<tr>
<td>{ ... }</td>
<td>2, 4</td>
<td>1</td>
<td>6</td>
<td>1</td>
</tr>
<tr>
<td>{ ... }</td>
<td>3</td>
<td>1</td>
<td>7</td>
<td>1</td>
</tr>
<tr>
<td>4 of {Dell, Desktop, Computers, OptiFlex, line, introduction, high, performance, Pentium, microprocessors,</td>
<td>0, 1</td>
<td>5</td>
<td>5</td>
<td>0</td>
</tr>
<tr>
<td>Query</td>
<td>F</td>
<td>P</td>
<td>R</td>
<td>E</td>
</tr>
<tr>
<td>----------------------------------------------------------------------</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td><code>computer, price, barrier, configured, system, speed, megahertz, Intel, chip, Doug, MacGregor, chips, machines, 486, models, power, management, networking, plug, play</code></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>{ ... }</code></td>
<td>2</td>
<td>1</td>
<td>7</td>
<td>2</td>
</tr>
<tr>
<td><code>{ ... }</code></td>
<td>3</td>
<td>1</td>
<td>7</td>
<td>1</td>
</tr>
<tr>
<td><code>{ ... }</code></td>
<td>4</td>
<td>1</td>
<td>6</td>
<td>1</td>
</tr>
<tr>
<td>5 of <code>{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}</code></td>
<td>0</td>
<td>6</td>
<td>6</td>
<td>0</td>
</tr>
<tr>
<td><code>{ ... }</code></td>
<td>1</td>
<td>4</td>
<td>7</td>
<td>1</td>
</tr>
<tr>
<td><code>{ ... }</code></td>
<td>2, 4</td>
<td>1</td>
<td>6</td>
<td>1</td>
</tr>
<tr>
<td><code>{ ... }</code></td>
<td>3</td>
<td>1</td>
<td>7</td>
<td>1</td>
</tr>
<tr>
<td>6 of <code>{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}</code></td>
<td>0-2</td>
<td>1</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td><code>{ ... }</code></td>
<td>3</td>
<td>1</td>
<td>7</td>
<td>1</td>
</tr>
<tr>
<td><code>{ ... }</code></td>
<td>4</td>
<td>1</td>
<td>6</td>
<td>1</td>
</tr>
<tr>
<td>2 of <code>{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}}</code></td>
<td>0-4</td>
<td>4</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td><code>{ ... }</code></td>
<td>1</td>
<td>4</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td><code>{ ... }</code></td>
<td>3</td>
<td>6</td>
<td>6</td>
<td>0</td>
</tr>
<tr>
<td><code>{ ... }</code></td>
<td>4</td>
<td>5</td>
<td>7</td>
<td>1</td>
</tr>
<tr>
<td><code>{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}}</code></td>
<td>0-4</td>
<td>2</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td><code>{ ... }</code></td>
<td>0</td>
<td>6</td>
<td>6</td>
<td>0</td>
</tr>
<tr>
<td><code>{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}}</code></td>
<td>0-4</td>
<td>2</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td><code>{ ... }</code></td>
<td>1</td>
<td>3</td>
<td>10</td>
<td>3</td>
</tr>
<tr>
<td><code>{ ... }</code></td>
<td>2</td>
<td>0</td>
<td>15</td>
<td>7</td>
</tr>
<tr>
<td><code>{ ... }</code></td>
<td>3</td>
<td>1</td>
<td>16</td>
<td>7</td>
</tr>
<tr>
<td><code>{ ... }</code></td>
<td>4</td>
<td>1</td>
<td>17</td>
<td>6</td>
</tr>
</tbody>
</table>
Index

A
Accessibility .................................. 12
Accuracy ....................................... 23
Adaptive resonance theory .................. 9
Application properties
  Flexibility .................................... 12
Artificial neural networks .................. 8
  properties .................................... 8
Artificial query ................................ 21
Automatic feature extraction .............. 9
B
Best matching unit ............................ 9
C
Context size .................................. 22; 50
Correlation .................................... 11
  Levels of .................................... 23
Correlation calculation
  Average error ................................ 23
  Average hit error ............................. 24
  Perfect hit rate ............................... 24
D
Data (*.dat) .................................. 17
Data set ....................................... 20
Database environment
  Dynamic ...................................... 7
  Static ........................................ 6
Demo (*.dem) .................................. 17
Demo dialogue ................................ 55
Demo mechanism ............................. 17
Distributed data representation .......... 8
E
Epsilon end .................................... 51
Epsilon start ................................... 51
Euclidean distance ........................... 9
Evaluation mode .............................. 19
Evaluation mode dialogue .................. 56
F
Files and Paths dialogue .................... 53
Filter principle ................................ v
Full-text query ................................ 21
Fuzzy search ................................... 7
G
Generalisation factor ......................... 22
H
Hashing ........................................ 14
Hints dialogue ................................ 54
Hit threshold ................................... 22; 51
Hitlist (*.hit) ................................... 17
Hooverhelp ..................................... 49
I
Index ............................................. 7
Information retrieval ......................... 6
  Drawbacks ................................... 7
  Index-based approach ....................... 6
Information storage sources ............... v
Information value ............................. 23
Input fibres .................................... 8
Internal data flow ............................ 52
K
k-d tree ........................................ 14
Kohonen feature map ......................... 8
  Properties ................................... 9
L
Language independency ...................... 8
M
Maximum vector spread ...................... 54
Measurement of results ..................... 70
N
Nearest-neighbour search ................... 13
Neural filter algorithm ...................... 10
Neuron ......................................... 8
n-gram ......................................... 10
n-gram analysis method ..................... 10
O
Object-oriented programming ............... 38
Open database connectivity ................ 17
Operators
  Conjunction ................................... 7
  Disjunction ................................... 7
  Negation ...................................... 7
  Proximity ..................................... 7
  Quorum ....................................... 7
Output (*.out / *.txt) ....................... 17
P
Parameters dialogue ......................... 50
Performance ................................... 12
Picture (*.pic) .................................. 17
Precision ....................................... 23
Preferences dialogue ....................... 52
Pre-processing phase ....................... 18
Q
Query
  Dynamic database environment .......... 7
  Operators .................................... 7
  Static database environment ............. 6
<table>
<thead>
<tr>
<th>Letter</th>
<th>Term</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>R</td>
<td>Recall</td>
<td>23</td>
</tr>
<tr>
<td></td>
<td>Robust behaviour</td>
<td>8</td>
</tr>
<tr>
<td>S</td>
<td>Selective Dissemination of Information</td>
<td>v</td>
</tr>
<tr>
<td></td>
<td>Self-organisation</td>
<td>9</td>
</tr>
<tr>
<td></td>
<td>Sensor</td>
<td>8; 50</td>
</tr>
<tr>
<td></td>
<td>Settings (*.set)</td>
<td>17</td>
</tr>
<tr>
<td></td>
<td>Sigma end</td>
<td>51</td>
</tr>
<tr>
<td></td>
<td>Sigma start</td>
<td>51</td>
</tr>
<tr>
<td></td>
<td>Simple recurrent network</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td>Sort hitlist dialogue</td>
<td>56</td>
</tr>
<tr>
<td></td>
<td>Space as character</td>
<td>22</td>
</tr>
<tr>
<td></td>
<td>State of self-organisation</td>
<td>43</td>
</tr>
<tr>
<td>T</td>
<td>Table (*.tbl)</td>
<td>17</td>
</tr>
<tr>
<td></td>
<td>Table map</td>
<td>15</td>
</tr>
<tr>
<td>U</td>
<td>Unsupervised training</td>
<td>9</td>
</tr>
<tr>
<td>V</td>
<td>Vector</td>
<td>8</td>
</tr>
<tr>
<td></td>
<td>View threshold</td>
<td>51</td>
</tr>
<tr>
<td></td>
<td>Visualisation</td>
<td>12</td>
</tr>
<tr>
<td></td>
<td>Textual</td>
<td>15</td>
</tr>
<tr>
<td>Z</td>
<td>ZyIMAGE</td>
<td>29</td>
</tr>
<tr>
<td></td>
<td>Query</td>
<td>79</td>
</tr>
</tbody>
</table>