Institute for Language, Logic and Information

# NEURAL NETS AND THEIR RELEVANCE FOR INFORMATION RETRIEVAL

J.C. Scholtes

ITLI Prepublication Series
for Computational Linguistics CL-91-02

University of Amsterdam

Instituut voor Taal, Logica en Informatie
Institute for Language, Logic and Information

Faculteit der Wiskunde en Informatica
(Department of Mathematics and Computer Science)
Plantage Muidergracht 24
1018TV Amsterdam

Faculteit der Wijsbegeerte
(Department of Philosophy)
Nieuwe Doelenstraat 15
1012CP Amsterdam

# NEURAL NETS AND THEIR RELEVANCE FOR INFORMATION RETRIEVAL

J.C. Scholtes
Department of Compuational Linguistics, Faculty of Arts
University of Amsterdam
email: scholtes@alf.let.uva.nl
mail: Dufaystr.1, 1075GR Amsterdam
fax: +31 20 6710793

*ITLI Prepublications*
*for Computational Linguistics*

Received October 1991

This research is supported by
MSC Beheer BV, Amsterdam

*Abstract*

*This paper presents two types of implemented neural methods for free-text data-base search. In the first method, a specific interest (or "query") is taught to a Kohonen feature map. By using this network as a neural filter on a dynamic free-text data base, only associated subjects are selected from this data base. The second method can be used in a more static environment. Statistical properties (n-gram or keyword distributions) from various texts are taught to a feature map. A comparison of a query with this feature map results in the selection of texts closely related to each other with respect to their contents.*

*All methods are compared with classical statistical information-retrieval algorithms. Various simulations show that the neural net indeed converges towards a proper representation of the query as well as the objects in the data base. The first algorithm seems much better scalable (linear versus exponential complexity) than its statistical counterparts, resulting in higher speeds, less memory needs, and easier maintainability. The second one particularly shows an elegant and uniform generalization and association method, increasing the selection quality.*

*By combining research results from connectionist Natural Language Processing (NLP) and Information Retrieval (IR), a better understanding of neural nets in NLP, a clearer view of the relation between neural nets and statistical Pattern Recognition, and an increased Information Retrieval quality are obtained.*

*Keywords: Information Retrieval, Neural Nets, Kohonen Self-Organizing Feature Maps, Natural Language Processing, Statistical Pattern Recognition.*

## Background

The Information Retrieval (IR) problem has many facets. The queries as well as the data base elements may be characterized by either static or dynamic features. Information filtering relates to static queries in a dynamic data-base environment. Here, one teaches a common interest to a filtering device, which selects interesting free-text with respect to the filter. Regular free-text search refers to a more static data base with dynamic queries. Due to the static character of the data base, elements can be preprocessed. In the retrieval phase, one compares the statistic analysis of a query with all the analyses of elements in the data base. Highly correlated analyses suggest a common subject [Croft et al., 1979].

The level of analysis in IR varies between statistical pattern recognition and a symbolic linguistic approach. Clearly, the retrieval quality dependents heavily on the amount of context and conceptual knowledge that is available in the retrieval phase. However, linguistic approaches result in complicated and computationally complex systems that are not quite usable in practical implementations. On the other hand, statistical pattern recognition techniques are quite unable to handle conceptual relations and higher order grammatical inferences, which are important to get the retrieval quality above the level of global surface analyses.
Generally, IR systems use statistical matching methods on either characters or words. Context is mostly represented by Markov chains on characters or words. Normally, the analysis of meaning doesn't go beyond the usage of synonyms [Rijsbergen, 1979], [Lancester, 1979], [Salton, 1968, 1971, 1980, 1986, 1989].

The free-text search problem (design and implementation of an efficient query system for a large unformatted text data base) can be approached with various techniques [Tenopir, 1984], [Barrett, 1989], [Tenopir et al., 1990]. An obvious search method is a keyword matching algorithm between a query and keyword records of separated parts in the data base (i.e., papers, or stories). The disadvantage of these algorithms is that one must either preprocess the text (attaching keywords to a data base object) or search all texts for a single query. The first problem can be solved by automatic indexing algorithms (which might be useful in a static environment, but are completely useless for the filtering problem) [Salton et al., 1968, 1973], [Sparck Jones, 1971], [Salton, 1972], [Willett, 1979].

[Stanfill et al., 1986,1989] propose an efficient method for the filter problem. Here, a massively-parallel free-text search method is implemented on the Connection Machine. This matching algorithm is probably the most thorough one possible, but it is quite expensive due to the need for parallel hardware [Pogue et al., 1987], [Salton et al., 1988], [Weyer, 1989], [Waltz, 1990], [Frieder et al., 1991], [Oddy et al., 1991].

Single keyword string-matching algorithms often result in an enormous amount of possible data base objects. By incorporating context in the queries and retrieval functions, irrelevant information can be eliminated from the retrieval set. Some derive context dependencies from boolean relations between keyword occurrences [Salton et al., 1983]. However, besides the increasing complexity in retrieval evaluation functions it is hard to express a query in such boolean relations. In short: single keywords often result in an over-kill in information, boolean keywords mostly cause an under-kill.

Another method to do text recognition is the n-gram search algorithm. This mechanism can be implemented on machines less powerful than the Connection Machine, but it still provides enough distinction between different objects in the data base. A query (one or more sentences) is reduced to a n-gram vector, representing the most frequent *n* character combinations in the query. This vector is compared to n-gram vectors of texts in the data base. By shifting a window over the text, relations between words are recorded without the need for a dictionary, prefix- and suffix stripping and boolean relations. The method has shown to be very successful in information retrieval problems. However, there are some major drawbacks.

First, there is the Markovian nature of the model. It cannot remember strings longer than the order of the Markov chain, even when a larger context is relevant to distinguishing two objects. One can extend the order of the chain, but every step results either in exponential memory usage or in the exponential increase of computation time. So, the n-gram method is not really scalable to higher order dependencies (e.g., 5-Gram word chains).
Second, the implementation of higher order n-grams requires skilled programming techniques. I.e., the statistical tables should be hashed, ordered, and normalized. Because of the trade off between memory and speed, one optimizes differently for different orders of the problem. So no uniform method can be used.
Third, there is still no meaning involved in the comprehension method; only structural features of the text are taken in account. [2]

As mentioned before, statistical IR methods have some shortcomings. The keyword based methods need a dictionary and suffix stripping algorithms. If one uses n-gram methods to eliminate the need for a dictionary, the complexity of the problem is exponential with respect to the window size. Furthermore, the window size limits the memory length. Moreover, the incorporation of meaning, other than synonyms, is hard to carry out with a statistical method. Finally, although statistical methods (computer procedures) provide a flexible method to implement local optimizations, all the normalizations, orderings, generalizations and associations *must* be programmed explicitly.

Especially this last reason combined with the need for longer memory (scalability) and possibilities to attach some (implicit) mechanism for meaning determination, resulted in the research described here. Research in neural nets showed good results in other pattern recognition tasks. Implicit parallelism, easy incorporation of knowledge from different sources, good generalization and easy association capabilities are the best known examples of advantages of neural nets. So why not use them for another classification task: *Information Retrieval.*

IR needs context. Recent research in connectionist Natural Language Processing (NLP) showed interesting results in self-learning systems [Elman, 1988], [Scholtes, 1991a-c]. The proposed models can learn Regular (Finite State) Grammars from unformatted sentences by using an (infinite length) Markov chain on words. In IR, this problem can be simplified to the use of finite length conditional probabilities over characters and words. Other research shows automatic categorizations of unknown words into appropriate clusters [Ritter et al., 1989b], [Elman, 1988], [Scholtes, 1991a-c]. Such automatic derivation of synonyms and related objects might be used to incorporate a simple notion of meaning in IR.

Although these methods are not capable to analyse complex linguistic structures, they do distinguish different contents better than global surface analyses, while they are still based on fast and automatically derivable learning and retrieval algorithms.

---

[2] We are aware that there are many possibilities to optimize the statistical methods. Much has been written on the string matching problem, clustering algorithms for n-gram vectors, etc. All resulting in better solutions for the brute force methods as proposed here. On the other hand, we don't incorporate these methods in our comparison because neural nets can be optimized in the same way with the same results [Kelly, 1991], [Koikkalainen et al., 1990]. Therefore, only plain, non-optimized methods are compared.

On of the first efforts to use connectionist methods in information retrieval can be found in [Mozer, 1984]. This was indeed a very localist solution, which mainly uses the parallel and generalization characteristics of PDP systems. Continuing this line of research resulted in [Belew, 1986], [Belew, 1987], [Bein et al., 1988], which uses localist connectionist systems to build networks of concepts. More recent work incorporates neural and symbolic techniques in information retrieval: [Belew et al., 1988], [Belew, 1989], [Rose et al., 1989a-b], [Rose, 1990], [Rose et al., 1991], [Rose, 1991]. The possible application of standard information retrieval strategies in (localist) neural nets is showed in [Wilkinson et al., 1991]. A good overview article of various efforts in connectionist information retrieval can be found in [Doszkocs et al., 1990]. These references describe the most important early work in connectionist IR.

Recently, a whole series of new papers appeared, reporting the use of Back Propagation (BP), Simple Recurrent Nets (SRN), Hopfield Nets, and Kohonen Feature Maps (KFM) in information retrieval. [Gersho et al., 1990a-b] propose a multi-layer hybrid neural net system. A Kohonen Feature map is used to determine global data clusters, while various backpropagation networks are taught to classify specific elements into these common clusters. The system is tested on a real data base and results in 93% correct retrievals. The main advantage of the neural method above the statistical and structural ones was the very short development time of the system. [Lin, 1991] used a Kohonen feature map to cluster 140 Artificial Intelligence papers based on the use of 25 keywords in the paper titles. By using this method, related papers cluster in neighbouring regions on the feature map.

[Allen, 1991] uses a Simple Recurrent Network (SRN) to answer questions on semantic aspects of simple propositions. Others teach bigram vectors to a regular back-propagation network [Mitzmann, 1991], or derive library categories with an SRN from book titles [Wermter, 1991]. In [Jagota, 1990a-b], [Jagota et al., 1990] and [Jagota, 1991] a Hopfield net is used to derive and store a large lexicon.

The research reported on in the current paper builds on what's good from the n-gram methods and has developed a Neural Filtering mechanism for (dynamic) free-text data bases. The statistical as well as the neural algorithms have been implemented and are compared to each other. Where the statistical method is fast for small dimensions of the problem, the neural (Kohonen) feature map shows considerable advantages for higher orders in as well speed, memory need, scalability, implementation ease, generalization and selection power. Implemented in or with parallel hardware, the neural method definitely outperforms the statistical.

If the data base is a more static one, objects can be clustered on (predetermined) keywords, abstracts or even on the entire text. By clustering such related objects, it is easier to discover correlated objects. One can determine the best group instead of the best paper (factor fewer comparisons). In neural nets research, such clusters are formed on so-called feature maps. If the neuron that correlates best to the query is found, the paper represented by this neuron and all papers in neighbouring regions are probably of the same category.[3]

On the one hand, connectionist NLP techniques can increase the retrieval quality. On the other hand, the IR problem can contribute to the understanding of neural nets as pattern classifiers by comparing neural information retrieval with (already well known) statistical information retrieval results.

[3] Plain Kohonen feature maps have the disadvantage that one has to take into account the Euclidean distance as well as the cluster boundaries if a measure of correlation between two objects is determined. More advanced methods, which automatically develop a feature map that fits the underlaying probability distribution better are under consideration [Fritzke, 1991a,b], [Martinetz et al., 1991].

## Introduction

A statistical algorithm that also incorporates some context and that can be used to implement synonyms is the n-gram vector method. Formally, an n-gram is an $n^{th}$ order Markov chain over character strings. Less formally it can be described as an n-length sequence of characters occurring in a word. For example, the trigrams ($n = 3$) occurring in the word *trigram* are *--t, -tr, tri, rig, igr, gra, ram, am-, m--* (the - indicates a space). An n-gram frequency vector can be viewed as a document finger print, documents can be identified by such vectors.
Normally, 2-grams are not distinguishing enough, trigrams (3-grams) yield enough distinction and can be practically calculated, 4-grams do not add more difference in feature vectors, worth the computational power, 5-grams are almost impossible to calculate and resemble keyword vectors. N-gram vectors provide enough distinguishing power only then if common words and common endings are eliminated from the text learned to the neural map. Furthermore, by multiplying n-gram frequencies with weight values (high values for rare n-grams and low values for frequent n-grams), less frequent n-grams may be accentuated. Synonym tables can create the illusion of intelligent behaviour at a reasonable (computational) price. In short: n-gram vectors are very powerful, easily manipulable, self-learning and language independent [4] [Forney, 1973], [Hanson, 1974], [Neuhoff, 1975], [Shingal et al., 1979a-b], [Hull et al., 1982], [Shihari et al., 1983], [Shihari, 1985], [D'Amore et al., 1988], [Kimbrell, 1988].

The first neural efforts in information retrieval based on localist and backpropagating neural nets showed considerable advantages over regular IR techniques. However, the so-called cluster network types are more suited for the IR task (as indicated by [Honkela et al., 1991] and shown by [Gersho et al., 1990a,b]). These models can be used to derive clusters from unformatted input data by using an unsupervised learning algorithm. The Kohonen network is known to implement a vector quantization algorithm, well suited for clustering purposes.

The Kohonen formalism is a competitive learning algorithm [Kohonen, 1982a-c, 1984, 1988, 1990a-b]. A two-dimensional map is constructed in a rectangular or hexagonal structure from individual neurons. Each neuron has a number of input sensors with an input activation and an input weight. All neurons have the same number of input sensors. The learning rule acts in the following way. First, copy the activation values of an input element into all input activation sensors of all neurons. Next, determine the best match by finding the neuron with the minimum (e.g., Euclidean or Cosine) mathematical distance between input and weight values. Then, adapt the weights of the neurons within a certain region of this minimum, so they'll recognize the current input vector better in the future. After numerous cycles, a topological map is formed, holding related elements in neighbouring regions.

To cut the noise and to restrict the input space, some measurements ought be taken. First, all lower case characters should be transformed to upper case. Furthermore, all non-alphabetic characters must be eliminated (digits, point, comma's, etc.).

For reasons of efficiency, all irrelevant n-grams have to be eliminated so the rare ones are accentuated. Therefore, one has to remove non-relevant words within a language (e.g., the, a, an, all, every, who, which, etc.). This might sound awful for a psycholinguist, but one should remember that this solution treats information filtering as a pattern recognition problem. Next, eliminate all common word endings such as: -ing, -ant, -end, etc. The remaining n-grams can be taught to the feature map in order of appearance according to the Kohonen formalism [5].

---

[4] Although the quality of retrieval is increased by eliminating specific words and word-endings (which are in fact language dependent), this method is still categorized as being language independent because this is just a simple (very trivial) table of words. Normally, this list isn't longer than 250 words and about 20 endings (see also the next footnote).

[5] If enough data for a specific field is available, the detection of frequent (or non-relevant) words can also be done automatically by a preprocessor. However, every language has its own non-relevant words, these hold for all different corpera. Therefore, the perfect filter set would contain the domain dependent as well as the domain-independent word set.

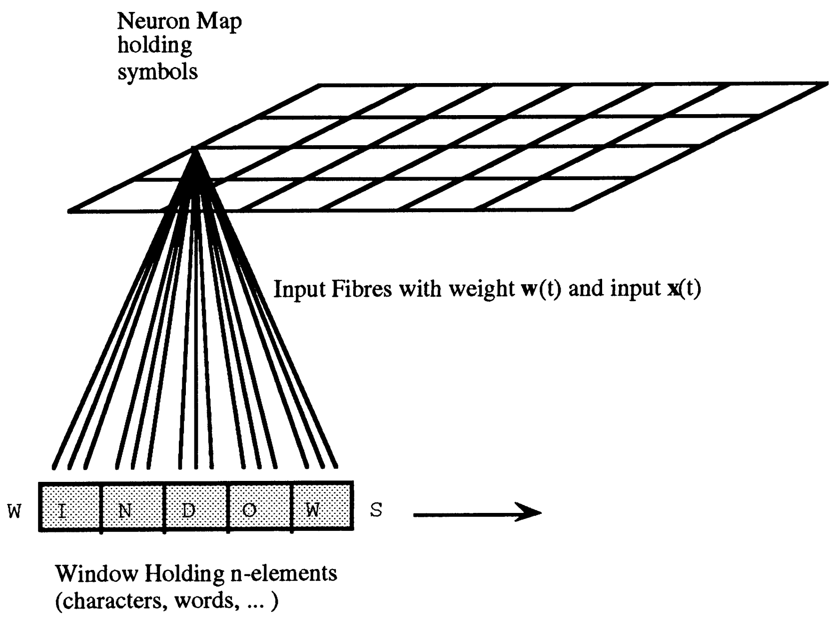
## Models and Algorithms

This paragraph discusses the models and corresponding algorithms in detail. The models are based on the Kohonen learning rule and on extensions of this model.

*Algorithm 1.1: The Kohonen Neural Filter Based on Characters*

The n-gram analysis method can be interpreted as a window size *n*, shifting over the words. This can be implemented quite simply in the Kohonen input sensors by assigning several sensors to each element in the window and concatenating all the window sensors to one big input vector. By shifting this window over the learning text, only frequent n-grams form clusters on the feature map, the others are overruled.

After learning, texts corresponding best to the query in the feature map will fit best to the clusters in the map (i.e., will yield the lowest cumulative error). Thus, this type of feature map can be used as a filtering device in an environment with a static query and a dynamic information flow. The method can be extended by incorporating spaces, so it learns simple contextual and semantical relations between words.




Algorithm 1.1: Neural Filter with N-Grams

**Step A: Teach Query to a Neural Net**

*0. Initialize and determine best learning parameters*
*1. Change all lower case characters to upper case*
*2. Eliminate all non-alphabetic characters*
*3. Eliminate non-relevant words*
*4. Eliminate non-relevant word endings*
*5. Determine n-grams*
*6. Teach trigrams to Kohonen feature map*

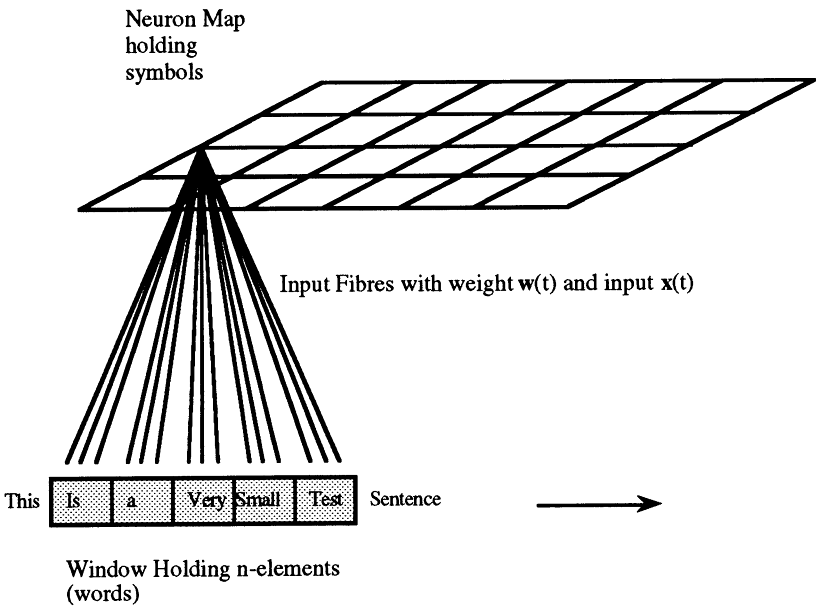
**Step B: Pass Free Text along Neural Filter**

*0. Determine text start-end (line, passage, paragraph, separator, etc.)*
*1. Change all lower case characters to upper case*
*2. Eliminate all non-alphabetic characters*
*3. Eliminate non-relevant words*
*4. Eliminate non-relevant word endings*
*5. Determine n-grams*
*6. Input n-grams to neural net and determine error*
*7. Select text if cumulative error < threshold*

*Algorithm 1.2: The Neural Filter Based on Words*

In the second neural filtering algorithm, the system has access to a small dictionary of 500 to 1,000 words. Every word has a unique code of some sensor values. After elimination of non-relevant words (words that are not in the lookup table) and word-endings, a vector representing a Markov chain over words is calculated. This vector is taught to the system. After passing the learning text multiple times, the Kohonen feature map represents a representation of common word combinations in the learning text.

By processing the retrieval text similarly, the retrieval algorithm incorporates contextual relations. The measure of correlation between these vectors and the representation on the feature map, determines whether a text part can be selected or not.
In this example all words are taught to the net. However, sometimes a word does not occur in the dictionary (because it is irrelevant for the selection process). The model ignores these words. As a result, it determines context from the relations between the remaining words.




Algorithm 1.2: Neural Filter with a Markov Chain on Words

**Step A: Teach Query to a Neural Net**

0. *Initialize and determine best learning parameters*
1. *Change all lower case characters to upper case*
2. *Eliminate all non-alphabetic characters*
3. *Eliminate non-relevant words*
4. *Eliminate non-relevant word endings*
5. *Determine vector representing Markov chain on known words*
6. *Teach word n-grams to Kohonen feature map*

**Step B: Match Free Text with Neural Filter**

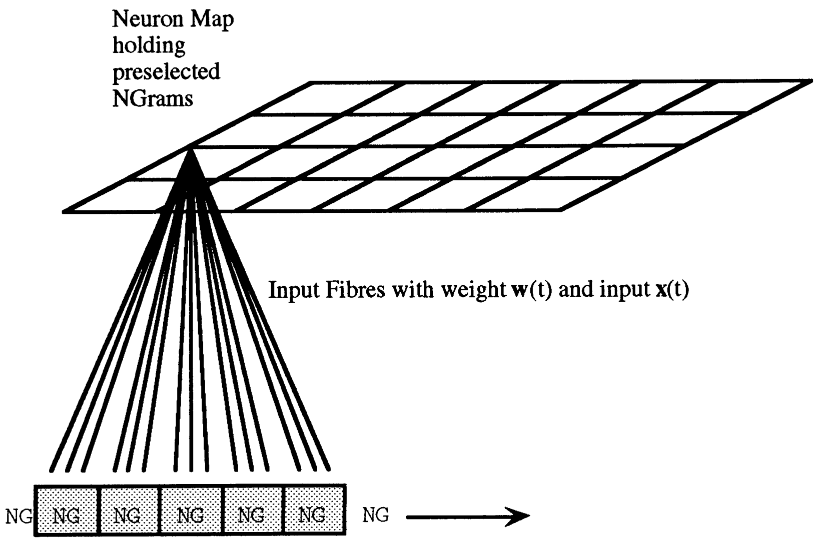
0. *Determine text start-end (line, passage, paragraph, separator, etc.)*
1. *Change all lower case characters to upper case*
2. *Eliminate all non-alphabetic characters*
3. *Eliminate non-relevant words*
4. *Eliminate non-relevant word endings*
5. *Determine vector representing Markov chain on known words*
6. *Input this vector to the neural net and determine error*
7. *Select text if cumulative error < threshold*

*Algorithm 1.3: The Neural Filter Based on (Preselected Large) N-Grams*

Sometimes the vocabulary of a specific domain is not known exhaustively or it is very dynamic. In such a case the model first calculates the most frequent n-grams (for large *n ≈ average word length in a language*). Then all non-relevant n-grams are eliminated from the learning text.

By shifting a window over the remaining n-grams, the neural map learns a representation of these n-gram combinations.

If the number of n-grams exceeds the addressing space, more n-grams might be eliminated from the learning text manually or on the basis of frequency.




Window Holding n-elements: Preselected Large NGrams

Algorithm 1.3: Neural Filter with a Markov Chain on (Preselected Large) N-Grams

**Step A: Teach Query to a Neural Net**

*0. Initialize and determine best learning parameters*
*1. Change all lower case characters to upper case*
*2. Eliminate all non-alphabetic characters*
*3. Eliminate non-relevant words*
*4. Eliminate non-relevant word endings*
*5. Determine n-grams*
*6. Select known n-grams*
*7. Determine vector representing the Markov chain of such known large n-grams*
*8. Teach Vectors to Kohonen feature map*

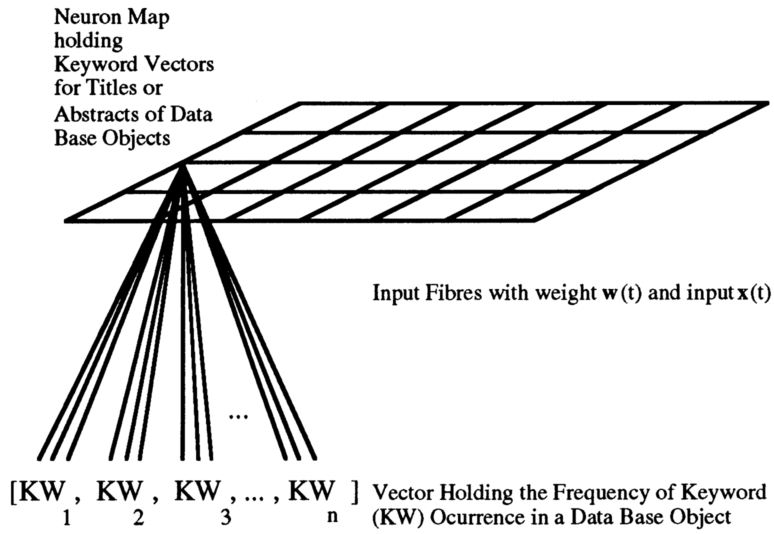
**Step B: Match Free Text with Neural Filter**

*0. Determine text start-end (line, passage, paragraph, separator, etc.)*
*1. Change all lower case characters to upper case*
*2. Eliminate all non-alphabetic characters*
*3. Eliminate non-relevant words*
*4. Eliminate non-relevant word endings*
*5. Determine n-grams*
*6. Select known n-grams*
*7. Determine vector representing the Markov chain of such known large n-grams*
*8. Input vectors to neural map and determine error.*
*9. Select text if cumulative error < threshold*

*Algorithm 2.1: The Neural Interest Map Based on Keyword Clustering*

The algorithms just mentioned use the Kohonen feature maps in a way they were not meant to be. They only remember a group of n-grams or keywords which occur frequently within a certain context (globally these n-grams are not necessarily the most frequent ones!). The resulting topology of the feature map is ignored completely (See the sections *Results* and *Discussion* for a more thorough deliberation on this subject).

Another, more normal use of feature maps is clustering of keywords that represent interests. Assume a full-text data base and a limited vocabulary in a specific domain (about 1,000 words). Then, each object can be represented by a vector holding a dimension for the frequency of every keyword. By teaching the keyword vector for every data base object to the Kohonen feature map, a topological representation of various interests will occur. Such a map might be seen as a neural interest map, where related papers are clustered in adjacent neighbourhoods.




The main difference between this method and work done by [Lin, 1991] is that this model uses the entire text (or that of an abstract) to cluster the papers, where Lin only uses 25 keywords occurring in paper titles. The amount of keywords used here is much larger (≈ 500). Moreover, the keywords are determined automatically by deriving the 500 most frequent (non-trivial) words in all the papers.

The map formed might be seen as a semantic map of the data base objects. Since [Doyle, 1961] there has been research towards the automatic formation of such maps. The author expressed his desire to use the computer not only as a tool in searching, but as a method to discover semantical relations. The approach taken by Doyle is quite simular to the neural net formalism of Kohonen. [Ritter et al., 1989b], [Ritter et al., 1990] and [Ritter, 1991] show possible applcation of such self-organizing *sematopical* maps in the derivation of semantic relations between regular words.

Moreover, there is a lot of literature on the functional specifications of a user friendly interface for document relations [Crouch, 1986]. The specifications pointed out in this work strongly resemble the characteristics of the Kohonen feature maps.

Although the relation between the cognitive and semantic maps as meant in the literature and the Kohonen formalism is not that direct, the Kohonen feature maps do share some properties of cognitive maps. Kohonen maps express relations between objects in euclidean distances, and they are able to reduce complex relations in an *n*- dimensional feature space into a lower two (or three) dimensional space with conservation of spatial and topological relations.

More on research toward the *cognitive* map can be found in [Lakoff, 1988], [Regier, 1988], [Chrisley, 1990], and [Palakal et al., 1991]

Algorithm 2.1: Interest Map Selected Keywords

**Step A: Teach Query to a Neural Net**

For all data base objects do:

1. *Change all lower case characters to upper case*
2. *Eliminate all non-alphabetic characters*
3. *Eliminate non-relevant words*
4. *Eliminate non-relevant word endings*
5. *Determine keyword vectors*

For all n-gram vectors do:

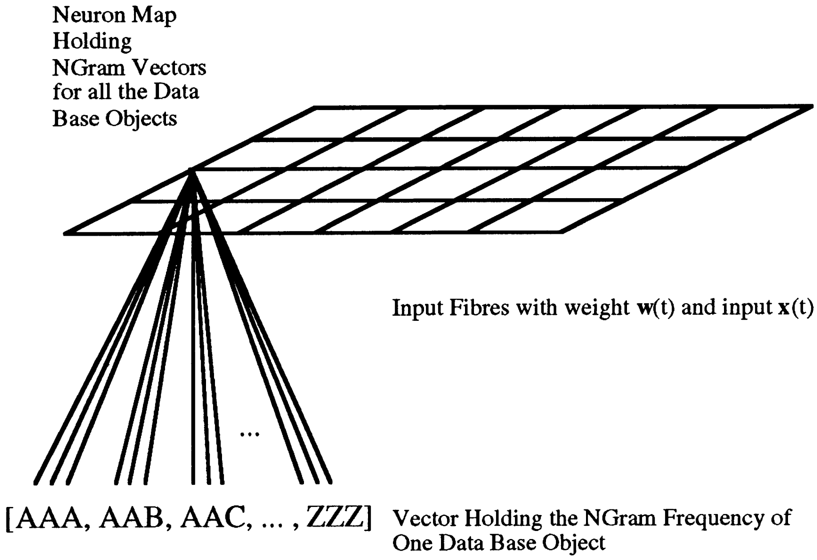
6. *Determine best learning parameters*
7.. *Code n-gram vectors*
8. *Reduce Dimension*
9. *Teach n-gram vector to the Kohonen feature map*

**Step B: Match Free Text with Neural Filter**

0. *Determine text start-end (line, paragraph, passage, section, separator, etc.)*
1. *Change all lower case characters to upper case*
2. *Eliminate all non-alphabetic characters*
3. *Eliminate non-relevant words*
4. *Eliminate non-relevant word endings*
5. *Determine selected keywords*
6. *Input selected keyword vector to neural net and determine error and position activated.*
7. *Select related data base objects if the error < threshold*

*Algorithm 2.2: The Neural Interest Map Based on N-Gram Clustering*

Algorithm 2.1 has the disadvantage of assuming a limited vocabulary that must be determined manually. The advantages of n-grams over keywords have been argued before. So, suppose we have calculated an n-gram vector holding the *n* most frequent n-grams for a certain (static) text stored in the data base. Then, a way of comparing a query to the database is by comparing the n-gram vector with all the n-gram vectors in the data base. If the data base holds many texts, this might be quite a job. Therefore, statistical methods use clustering algorithms and compare the n-gram vector to a cluster of data base elements. By learning the n-gram vectors of the data base to a Kohonen feature map, a topological (clustered) map of interest develops automatically, eliminating the need to program complicated clustering, generalization and association algorithms. To query the data base, a free-text query is processed like the learning text. The resulting vector is positioned on the map. By investigating the activity on the map, the area representing this vector can be found efficiently. All texts represented by neurons in the neighbouring region can then be considered of interest. A threshold function fine tunes the system.




However, these vectors have dimension $27^3 = 19{,}683$ (in the case of a 3rd order markov chain over 27 characters), which is definitely too much for any practical solution. But, we can transform this vector to a much smaller base, without losing too much of its characteristics. The legitimacy of this dimension reduction is based on the heuristic that most elements in the trigram vector: $\mathbf{y}(t)$ are about zero. Suppose the new basis consists of vectors $\zeta_i$. Then,a number of vectors $\zeta_i$ is generated randomly, one for each dimension of $\mathbf{y}(t)$. The reduced vector $\mathbf{y}'(t)$, is determined by computing the sum of the products of the separate dimensions of $\mathbf{y}(t)$ with the corresponding components of in $\zeta_i$.

$$\mathbf{y}'(t) = \Sigma_{i=1,N} ( y_i (t) \cdot \zeta_i )$$

where

| | | |
|---|---|---|
| N | = | Dimension vector $y_i$ (t) |
| $y_i$ (t) | = | Original vector with dimension N |
| $\zeta_i$ | = | Vector *i* from a set of N vectors of lower dimension *n* which altogether form the new basis for $\mathbf{y}'(t)$ |
| $\mathbf{y}'(t)$ | = | Transformed vector of lower dimension *n* |

By doing so, the number of fibres can be reduces enormously. So, even large maps learn complex representations within reasonable time limits. See [Ritter et al., 1989a] for a proof of the legitimacy of this operation.

After dimension reduction, a 500 up to 1000 dimensional vector remains, which represents all possible keyword relations without any dictionary and prefix- or suffix stripping.
Although this method still uses complicated algorithms to determine the initial n-gram vectors and to reduce them in dimension, the neural net smoothly solves the entire generalization and association process.

Algorithm 2.2: Interest Map Trigrams

**Step A: Teach n-gram vectors to a Neural Map**

For all data base objects do:

1. *Change all lower case characters to upper case*
2. *Eliminate all non-alphabetic characters*
3. *Eliminate non-relevant words*
4. *Eliminate non-relevant word endings*
5. *Determine n-gram tables, select the n most frequent elements*

For all n-gram vectors do:

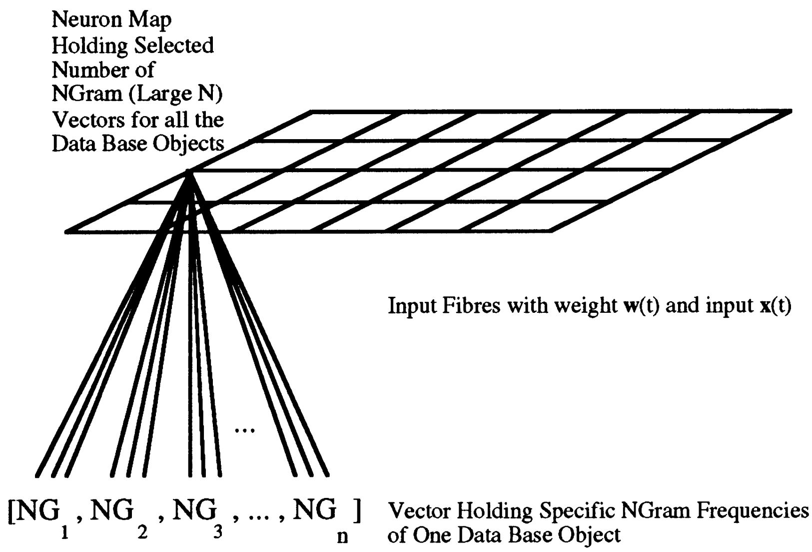
6. *Determine best learning parameters*
7.. *Code n-gram vectors*
8. *Reduce dimension*
9. *Teach n-gram vector to the Kohonen feature map*

**Step B: Match Query with Neural Map**

1. *Change all lower case characters to upper case*
2. *Eliminate all non-alphabetic characters*
3. *Eliminate non-relevant words*
4. *Eliminate non-relevant word endings*
5. *Determine n-gram vector, select the n most frequent elements*
6. *Reduce dimension*
7. *Input n-gram vector to neural net and determine error and position activated.*
8. *Select related data base objects if the error < threshold*

*Algorithm 2.3: The Neural Interest Map Based on (Preselected Large) N-Gram Clustering*

Now, what if the keywords are too limited (due to a dynamic and unknown vocabulary) and the trigram vectors end up having far less *zero* elements than we expected.




Then, we can derive the most frequent (large) n-grams and teach the Kohonen feature map a vector representing a specific n-gram in every dimension. After training, the feature map represents an interest map of the full-text data base. Objects related to each other in n-gram usage are within nearby clusters.

Algorithm 2.3: Interest Map Selected Large NGrams

**Step A: Teach n-gram vectors to a Neural Map**

For all data base objects do:

*1.* *Change all lower case characters to upper case*
*2.* *Eliminate all non-alphabetic characters*
*3.* *Eliminate non-relevant words*
*4.* *Eliminate non-relevant word endings*
*5.* *Determine n-gram tables, select the n most frequent elements*

For all n-gram vectors do:

*6a..* *Find best learning parameters*
*6b.* *Code n-gram vectors*
*7..* *Teach n-gram vector to the same Kohonen feature map*

**Step B: Match Query with Neural Map**

*1.* *Change all lower case characters to upper case*
*2.* *Eliminate all non-alphabetic characters*
*3.* *Eliminate non-relevant words*
*4.* *Eliminate non-relevant word endings*
*5.* *Determine n-gram vector, select the n most frequent elements*
*6.* *Input n-gram vector to neural net and determine error and position activated.*
*7.* *Select related data base objects if the error < threshold*

## Learning and Retrieval Rules

The learning rule used in the previous model is the Kohonen rule. It doesn't matter whether one uses characters, words or large n-grams as input elements, a coding procedure prepares all symbolic data for input to the feature map by translating them to vectors. This coding process is performed with the aid of a lookup table. All elements of the learning set are assigned randomly to specific codes in this lookup table. The codes itself are spread homogeneously through the feature space, to speed up the learning process. Convergence parameters as proposed by [Ritter et al., 1989a] fine tune the Kohonen rule.

### Kohonen Learning Rule

*0.*

| | | | |
|---|---|---|---|
| N | Number neurons in layer | n | Dimension of input vector |
| M | Set of all neurons in map | $\mathbf{w}(t)$ | Sensor weight |
| $\mathbf{x}(t)$ | Sensor input | | |

*1.* *Determine neuron s. This neuron has the net's best match between the input values and its weight values:*

$$\forall r \; \|\mathbf{w}_s(t) - \mathbf{x}(t)\| \leq \|\mathbf{w}_r(t) - \mathbf{x}(t)\| \qquad \text{for r element of M}$$

*2.* *Update all weights in the map according to the Kohonen learning rule:*

$$\mathbf{w}_r(t+1) = \mathbf{w}_r(t) + \varepsilon(t) \cdot \Phi_{rs} \cdot (\mathbf{x}(t) - \mathbf{w}_r(t))$$

*where:*

| | | | |
|---|---|---|---|
| $\Phi_{rs}$ | = | $e^{-(\|\| r - s \|\| / 2\sigma(t)^2)}$ | |
| $\varepsilon(t)$ | = | $\varepsilon_{max} \cdot (\varepsilon_{min}/\varepsilon_{max})^{t/t_{max}}$ | |
| $\sigma(t)$ | = | $\sigma_{max} \cdot (\sigma_{min}/\sigma_{max})^{t/t_{max}}$ | |
| $\varepsilon_{max}$ | $\in$ | [0,1] | Start Learning Rate |
| $\varepsilon_{min}$ | $\in$ | [0,1] | Final Learning Rate |
| $\sigma_{max}$ | = | $\sqrt{(N)} / 2$ | Start Region Size |
| $\sigma_{min}$ | $\in$ | [0,1] | Final Region Size |
| $\|\| r - s \|\|$ | = | Physical distance on map from neuron r to s | |

Once the feature map training is completed, we must match the test data with the representation formed on the neural map. In the case of the neural filter, one counts the cumulative (normalized) error or the cumulative (normalized) number of perfect hits (or whatever variant of these two functions, see box for details).

In general, one can separate two types of selection rules: positive and negative ones. The negative approach mainly filters the noise. A more positive approach is used to select possible candidates for selection. Negative selections are mostly normalized, while positive ones are not. If one paragraph in a paper is related to a specific interest, the positive filter selects it directly, where the negative one ignores the one paragaraph due to normalization of the retrieval value (one paragaraph fires high, all the others low, so the average firing level is still low). Positive selection mostly results in too many candidates where negative selection results in too few candidates. A proper combination of both approaches results in the best retrieval results.

Possible positive search methods are plain keyword matches and the (non-normalized) number of perfect hits on the neural map (in the case of n-gram on characters as well as n-gram on words). A negative filter is the added and normalized error of all text elements with respect to a statistical table or a neural map.

**Selection Rule 1 for Neural Filter (Negative)**

*1.* Select if: $((\Sigma_{\text{all n-grams in text part}} \|\mathbf{w}_s(t) - \mathbf{x}(t)\|)/\text{number n-grams in text part}) < \tau$

*where:*

| | | | |
|---|---|---|---|
| *s* has the property: | | $\forall r \; \|\mathbf{w}_s(t) - \mathbf{x}(t)\| \leq \|\mathbf{w}_r(t) - \mathbf{x}(t)\|$ | for r element of M |
| $\tau$ | $\in$ | [0,1] | Threshold Value |
| $\mathbf{x}(t)$ | = | vector holding one n-gram (direct coding through look up tables) | |

**Selection Rule 2 for Neural Filter (Positive)**

*2.* Select if: (Count(all n-grams in text part for which $\|\mathbf{w}_s(t) - \mathbf{x}(t)\| < \tau$)/number n-grams in text part) $> \phi$

*where:*

| | | | |
|---|---|---|---|
| *s* has the property: | | $\forall r \; \|\mathbf{w}_s(t) - \mathbf{x}(t)\| \leq \|\mathbf{w}_r(t) - \mathbf{x}(t)\|$ | for r element of M |
| $\tau$ | $\in$ | [0,1] | Threshold value before counting (very small) |
| $\phi$ | $\in$ | [0,1] | Threshold value before selection |
| $\mathbf{x}(t)$ | = | vector holding one n-gram (direct coding through look up tables) | |

*3.* *One might combine the rules 1 and 2 in an even more powerful mechanism.*

With the interest map, one determines the neuron representing the interest vector best and returns the paper represented by this neuron and all the other papers within the same cluster (determined by euclidean distance or by knowledge of the cluster boundaries on the feature map).

**Selection Rule for Keyword Neural Interest Map**

*1.* Select all objects represented by neighbouring neurons of neuron s

*s* has the property: $\forall r \ \|\mathbf{w}_s(t) - \mathbf{x}(t)\| \leq \|\mathbf{w}_r(t) - \mathbf{x}(t)\|$ and $\|\mathbf{w}_r(t) - \mathbf{x}(t)\| < \tau$ for r element of M

*where:*

| | | | |
|---|---|---|---|
| $\tau$ | $\in$ | [0,1] | Threshold Value |
| $\mathbf{x}(t)$ | = | vector holding keyword frequencies. One dimension for every keyword known by the system. | |

**Selection Rule for N-Gram Neural Interest Map**

*2.* Select all objects represented by neighbouring neurons of neuron s

*s* has the property: $\forall r \ \|\mathbf{w}_s(t) - \mathbf{x}(t)\| \leq \|\mathbf{w}_r(t) - \mathbf{x}(t)\|$ and $\|\mathbf{w}_r(t) - \mathbf{x}(t)\| < \tau$ for r element of M

*where:*

| | | | |
|---|---|---|---|
| $\tau$ | $\in$ | [0,1] | Threshold Value |
| $\mathbf{x}(t)$ | = | vector holding transformed (by dimension reduction) representation of n-gram frequencies. One dimension for every possible n-gram. | |

**Selection Rule for Selected (Large) N-Gram Neural Interest Map**

*3.* Select all objects represented by neighbouring neurons of neuron s

s : $\forall r \ \|\mathbf{w}_s(t) - \mathbf{x}(t)\| \leq \|\mathbf{w}_r(t) - \mathbf{x}(t)\|$ and $\|\mathbf{w}_r(t) - \mathbf{x}(t)\| < \tau$, for r element of M

*where:*

| | | | |
|---|---|---|---|
| $\tau$ | $\in$ | [0,1] | Threshold Value |
| r | = | All elements of Map M | |
| $\mathbf{x}(t)$ | = | vector holding frequencies of (preselected) n-gram occurrences. One dimension for every possible n-gram. | |

## Simulations and Results Neural Filter

The simulations are implemented on a high end PC (33 Mhz 386) and on a Sun Sparc Station IPC (the Neural Filter runs on the PC as well as on the Sun IPC. The Neural Interest Map runs only on the Sun IPC due to its huge memory requirements). The programs are written in C. The PC was connected to a CD-Rom player holding several free-text data bases such as: The Complete Works of Shakespeare, The Complete Sherlock Holmes, Microsoft Small Business Consultant and The Complete Translated 1987 Pravda articles. The algorithm was tested on selections of these CD-Roms. The most intriguing one was the Pravda data base, which shall be used in the examples of this paper.

The simulation parameters are determined automatically according to the best parameter heuristics in [Ritter et al., 1989b] and [Scholtes, 1991c]. Before training, the learning text was analysed to provide the optimal values for the internal coding, the learning rate, the region size, and the approximate number of necessary training cycles to reach the self-organizing state.

In the case of algorithm 1: the Neural Filter, the learning set holds a small selection on the 1987 nuclear weapon restriction talks between the USA and the USSR. Keywords as Reykjavik, ABM, Peace, etc. are more than once used. The test set was the entire Pravda CD-Rom (200 Mbyte), being passed along the neural filter.

Our era, a fast-paced era of nuclear weapons, an era of growing economic and political interdependence, precludes the possibility of security for one nation at the expense of others. I repeat: we can only survive or perish together. Security today can only be viewed as mutual, or to be more precise, universal. So whether we like each other or not, we need to learn how to coexist and live in peace on this small and very fragile planet. Question: Do you support the continuation in 1987 of the Geneva talks between Soviet and American representatives for the purpose of achieving progress on the issue of limiting and reducing arms? Answer: Yes, we do. We support talks that would overcome the state of fruitlessness and inertness and acquire true dynamism, in a word, talks that would become genuine talks on reducing nuclear arms and preventing an arms race in space. We tried to achieve that in Reykjavik and will try to achieve it even more energetically in 1987. I am sure that such a radical turnaround in the talks woul respond to the vital interests of the American people as well. At the same time, the position of the US administration on this issue is a cause of great disappointment for us. After Reykjavik the American delegation in Geneva has become even less cooperative. Despite the fact that the USSR has not been conducting nuclear detonations for 18 months, the USA has continued tests and refused to discuss a total ban on them, though it committed itself to conduct negotiations on that issue in the two treaties of 1963 and 1974. In November that was aggravated by the provocative action the White House took when it broke the important strategic arms limitaion agreement (SALT II). It does not help to conduct successful negotiations on new agreements when the old ones are being deliberately and blatantly broken. This is a serious problem that deserves very close attention. I will state once again that we support agreements on the most radical reductions of arms, both nuclear and conventional. Now it is up to Washington.

Learning Set (or 'Query')

Before learning, the text was filtered to avoid wasting computation time (or neural memory) to non relevant features. First all non-alphabetic characters (digits, read markers, etc.) were eliminated. Next all lower case characters were transformed to upper case. About 250 common English words were eliminated from the text. To obtain an even stronger distinguishing behaviour, all common endings (resulting in trigrams such as *-ary*, *-able*, and *-ent* ) were eliminated too. A number of such common words and word endings can be found in the tables below.

| | | |
|---|---|---|
| a | few | little |
| about | first | many |
| after | get | me |
| all | go | might |
| always | going | much |
| and | good | near |
| at | got | never |
| been | he | new |
| before | her | no |
| both | here | not |
| by | I | off |
| end | is | often |
| even | it | once |
| every | just | ... |

Part of Common Words Table

| |
|---|
| -ably |
| -ibly |
| -ily |
| -ss |
| -ous |
| -ies |
| -s |
| -ied |
| -ed |
| -ing |
| -... |

Part of Common Endings Table

*A First Statistical Analysis of the Learning Set*

A first analysis made was the determination of the 27 most frequent trigrams in the query by brute force counting. The three small tables below hold the values found. The first entity represents the trigram, the second the absolute frequency and the third the probability of occurrence, determined on basis of this text. One can right away spot the trigrams from words as *Weapons* and *Nuclear.* This small table is used for some global comparison between the statistical and neural methods. Off course, the text processed in this simulation was preprocessed according to the method presented in the boxes of algorithms.

| | | |
|---|---|---|
| AGR | 36 | 0.005380 |
| AIN | 36 | 0.005380 |
| ARM | 40 | 0.005977 |
| CLE | 52 | 0.007770 |
| CON | 48 | 0.007173 |
| DUC | 40 | 0.005977 |
| EAC | 40 | 0.005977 |
| EAR | 60 | 0.008966 |
| EAT | 44 | 0.006575 |

| | | |
|---|---|---|
| ENT | 84 | 0.012552 |
| ERI | 40 | 0.005977 |
| EST | 44 | 0.006575 |
| GRE | 48 | 0.007173 |
| INT | 44 | 0.006575 |
| ITY | 40 | 0.005977 |
| LEA | 56 | 0.008368 |
| MEN | 60 | 0.008966 |
| NUC | 48 | 0.007173 |

| | | |
|---|---|---|
| PEA | 36 | 0.005380 |
| PER | 36 | 0.005380 |
| POS | 36 | 0.005380 |
| PRO | 48 | 0.007173 |
| REA | 60 | 0.008966 |
| STR | 48 | 0.007173 |
| TER | 40 | 0.005977 |
| TRE | 36 | 0.005380 |
| UCL | 48 | 0.007173 |

Most Frequent Trigrams in Learning Text (Format: Trigram Frequency Probability)

*Results for the Neural Filter Algorithm Based on N-Grams*

The next simulation filtered the query and taught the trigrams (3-gram) up to 4 times to a 15 by 15 Kohonen feature map. Because the text contains many more trigrams than the 225 neurons can hold, only the most dominant (according to frequency and distance in internal coding) ones will be stored and each trigram is only held by one neuron.

The presented trigram map was obtained by determining the code for each neuron after the learning process had converged to a stable state. The meaning of the neuronal fibres can be looked up rapidly in the internal coding tables. The trigrams showed below are the *best* ones. It could well be that a specific neuron represents other trigrams as well within certain limits. By

comparing the most frequent trigrams obtained with the statistical method, one can observe that the trigrams represented by the neurons in the map are indeed within the most frequent.
The presented trigram map evolves by determining the code for each neuron after the learning process converged to a stable state. The trigrams for *NUCLEAR* are underlined as an example. In this one and all the following maps, *XXX* means no proper symbol could be found.

| | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ERN | MOM | LLY | PLA | HUM | ANI | UGG | TRY | LIM | TRA | STA | MPS | GTO | ENO | PRO |
| DOU | IMA | LAN | URT | REL | IMP | UPP | YME | <u>LEA</u> | LUD | KJA | TUA | YTH | ION | GIC |
| CUM | ANT | ITS | RAT | LIZ | SEE | TOU | ODW | MBE | EME | UTU | SSF | STR | DOF | POL |
| STS | MEN | ENT | <u>EAR</u> | REA | ICY | THE | ORY | DUC | ATE | ATY | APO | SUP | SOV | SEL |
| END | DIS | PHE | OSS | TAS | OAC | XIB | ELA | GRA | HIG | OLI | ARM | WID | ARY | ILI |
| GRE | ONT | UMS | LIT | UCT | <u>UCL</u> | REQ | TLY | BOD | ICA | QUI | NOB | ORI | GIL | FIN |
| ALL | EQU | BIL | ARC | EEP | UNC | OVO | CON | <u>CLE</u> | ISC | DET | OGR | OFF | RIT | ECI |
| AST | ALT | HIE | VOC | MPL | LAB | MAT | GLI | CLI | INT | RCH | RSE | OPO | RIZ | RAN |
| EST | OSP | UNR | PAC | IRE | OMP | TIP | OUS | NSI | ITI | OYM | INI | INC | IZO | UMA |
| RST | SSU | SHI | PEA | YST | NTE | OUR | NTA | BOR | NTO | ITY | CHI | EGI | OOD | OBO |
| NIT | USY | SET | EEM | PSE | ANG | ISA | EYK | AVI | NCL | CLU | BEC | HOR | HER | DAY |
| TIC | SPH | TUR | TIO | JAV | OPH | OVI | APP | ATO | SUC | RUG | <u>NUC</u> | VIK | DEE | ORT |
| TMO | TES | TSE | GER | SAR | MAI | MAL | ART | ANC | ILL | ZON | YKJ | xxx | xxx | AGR |
| ROP | FIC | SPL | VOR | EAR | MAN | REY | TON | LAT | ABO | NDO | DUR | NYT | IDE | OME |
| TEG | UIR | TER | UCI | GGR | ABM | AGA | EAC | TAN | ACE | ALI | MUT | GGL | GOT | BLY |

Trigram Feature Map of Query in Neural Filter

There are some aspects that are still not completely clear in this phase of the research. If we compare the most frequent trigrams (according to the statistics) with the trigrams in the feature map, then some of them are not there: *ain, eat, eri, per, pos, tre.* On the other hand, some trigrams occur that cannot be found in the learning text at all, such as: *ggl, xib.* This is probably due to the neighbourhood effects, which do indeed absorb less frequent trigrams, but they also disturb the internal coding of existing trigrams and cause the occurrence of non-existing trigrams.

These two phenomena are more clear in the following examples. Suppose we teach a single occurrence of the trigrams *aaa, aab, ..., aaz* and 10 trigrams *zzz,* and the Kohonen map consists of 1 neuron. Then, only *aaz* will be represented and not *zzz.* Although *zzz* is the most frequent one. But, the *aa...* combination occurs more frequently, so this effect can be defended as being positive (it does not remember the most frequent trigrams, but the most related). The other effect is more serious. Say we train many *aaa*'s and *aac*'s, then the model probably ends up in *aab* if it is constructed of one neuron, although *aab* never occurred! This effect explains the occurrence of non-existing trigrams in the feature map. However, the effect occurs only if the recognition threshold is too large. By sharpening the threshold, this effect might appear in theory, but practical emergence will be limited to rare cases. In fact, the trigrams *ggl* and *xib* in the above feature map had pretty large errors (>5%) [6].

Normally, it isn't worth the trouble to derive 4-, 5-, 6-, and 7-grams. The argument that trigrams give more than enough separation is used more than once. This argument holds if an increase of the window size corresponds to an exponential increase in space or time to derive such dependencies. However, here we have a linear increase of complexity as the window size grows (as shall be shown further on). This result makes it real easy and worth the trouble to derive such higher order n-grams and determine their influence on the retrieval process.

[6] I owe the discovery of these effects and the examples to Brain Bartell.

An advantage of neural nets over statistics follows undoubtedly from the following simulation. By increasing the window size from 3 to 4, the neural net learns 4-grams within almost the same amount of memory and at the same speed as the trigram implementation. However, the statistical method needs either a factor 27 more memory, or decreases dramatically in speed. One can recognize frequent words from the learning text even better than in the trigram map.

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| EREO | RUGG | LIMP | LITI | CAUS | PLOY | DISA | EOTY | GREA | TEND |
| ALIZ | WISH | EATY | LOYM | VIET | GLIM | ANYT | NUCL | GOOD | XTEN |
| TRIC | IGUR | COOP | SINC | WILL | ACHI | DSHI | SECU | ONTO | APPR |
| CURI | TOUR | OGRA | MPRO | TANC | ATEL | FRIE | SAPP | ABIL | AINT |
| EACE | EYKJ | RTUN | PROP | NCER | WEST | IZON | ENDS | LICY | ARCH |
| UCIN | SPAC | ABOR | VERN | NALL | OPOS | AVIK | ALLY | NCLU | OUSY |
| STAN | ISAR | HUMA | UNAT | STRI | EDUC | UCLE | REOT | AREA | SELF |
| TSEL | KJAV | CLEA | BORA | OSPH | MSTA | RIEN | NDSH | NMEN | CATA |
| UTUA | RELA | LIMA | REYK | ICAN | SHIP | EACT | RONG | MPLE | ATEG |
| TRUG | ASTR | TIPU | POLI | FIGU | OLIC | YKJA | ORIZ | MENT | LEAR |
| AGAI | SOVI | MOSP | OOPE | USSR | JAVI | OTYP | CLUD | IMPR | DOFF |
| IEND | SPHE | PSET | ITHD | ORTU | PAST | UBLY | FICI | EXTE | TUNA |
| POSS | ERNM | PEAC | UNFO | ESEA | INCE | OSIT | CUMS | MUTU | STIP |
| SCUS | BODY | EGIC | UGGL | REAC | IMPS | STRO | REAT | NTOU | AGRE |
| ISSU | REST | GOVE | ERST | PHER | OPER | GTON | OLIT | TLES | SUPP |

4-Gram Feature Map of Query in Neural Filter

The same holds for the 5-gram simulation: speed and memory requirements were about the same as for the trigram simulation. The possibility to represent almost 5-grams eliminates the need to develop a large dictionary of known words. 5-grams can be used in any language *without* a-priori knowledge.

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| STABI | TRICT | PERIT | ALLIS | REOTY | DERST | REATY | TSELF | SECUR | MAINT |
| VERYT | ALISM | TEGIC | POLIC | OLITI | PARTY | ANTLY | AGREE | POWER | JAVIK |
| ECURI | ORIZO | NFORT | CIRCU | NSIDE | OMPLE | STRUC | CURIT | ACEFU | ETELY |
| LITAR | MMEDI | REDUC | EACEF | NDOFF | NDSHI | ATEST | SCUSS | GUARD | UCLEA |
| EDUCT | WERFU | EREOT | EMAIN | FIRML | RDINA | UALIT | WEAKN | MPROV | RELAT |
| REMAI | COURT | BORAT | PURSU | SUPPO | SSARY | RUGGL | PLETE | PEACE | EFEND |
| ESSAR | RADIC | DEVEL | OWERF | XTRAO | SOVIE | QUALI | CESSA | ORTUN | EARCH |
| ISAPP | TUNAT | UMSTA | ANGER | EYKJA | URITY | DESTR | VELOP | THDRA | CLEAR |
| FRAGI | HUMAN | OPMEN | SINCE | TRUCT | ERFUL | TASTR | NEGOT | IENDS | EOTYP |
| ESTRU | CIALI | LABOR | DEFEN | ILITA | RTUNA | OBODY | TIPUL | IALIS | ATEGI |
| LITIC | XIBIL | COMPL | OVIET | EGOTI | MALIZ | ECESS | LIMIN | ARTIC | PMENT |
| QUIPP | GOODW | KJAVI | MPLET | YKJAV | AKNES | IRCUM | INCLU | BLATA | UMANI |
| PRINC | WEAPO | STRIC | VERNM | UCING | RMITT | XTEND | LIMPS | TMOSP | ONTOU |
| PURPO | HORIZ | SOCIA | APPRO | CLIMA | ERICA | EAPON | OCIAL | EVERY | IRMLY |
| DISPL | ERYTH | NECES | EQUIP | EVELO | EQUAL | STAND | PROPO | REYKJ | OSPHE |

5-Gram Feature Map of Query in Neural Filter

To convince the reader even more, the next two pictures hold n-gram feature maps for 6- and 7-grams. All calculated within the same amount of memory and at the same speed.

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| xxxxxxx | RIENDS | ROSPER | SECURI | xxxxxxx | COURSE | xxxxxxx | SOVIET | ECURIT | xxxxxxx | xxxxxxx | xxxxxx |
| NEGOTI | xxxxxxx | TREATY | ANDOFF | ONTOUR | OLITIC | OCIALI | LIMATE | xxxxxxx | xxxxxxx | WERFUL | EQUIPP |
| PARTIE | APPROA | SOCIAL | xxxxxxx | IALISM | CIRCUM | SEARCH | NDSHIP | ROGRAM | INCIPL | REDUCT | MSTANC |
| TURNAR | xxxxxxx | xxxxxxx | xxxxxxx | xxxxxxx | xxxxxxx | xxxxxxx | ORDINA | REYKJA | STRONG | INCLUD | ITSELF |
| POWERF | xxxxxxx | IMMEDI | xxxxxxx | ACEFUL | xxxxxxx | ATIONS | xxxxxxx | TRATEG | SPERIT | xxxxxxx | RMITTE |
| PEACEF | KJAVIK | DANGER | STEREO | xxxxxxx | POLITI | xxxxxxx | ODWILL | xxxxxxx | xxxxxxx | TROPHE | OWERFU |
| xxxxxxx | DISPLA | xxxxxxx | ACHIEV | ASTROP | xxxxxxx | DEVELO | xxxxxxx | ELOPME | xxxxxxx | QUALIT | PROPOS |
| CURITY | IRCUMS | xxxxxxx | NSIDER | HIGHER | xxxxxxx | MERICA | xxxxxxx | xxxxxxx | ISAPPO | YKJAVI | HORIZO |
| CUMSTA | xxxxxxx | RGETIC | xxxxxxx | TANTLY | REMAIN | NOBODY | DEFEND | EQUALI | VIOUSY | IENDSH | REOTYP |
| FIRMLY | SUPPOR | ECESSA | CONNEC | DEPLOY | ORTUNA | ENOUNC | PPROAC | HINGTO | UCLEAR | EVERYT | PPOINT |
| xxxxxxx | ATMOSP | xxxxxxx | xxxxxxx | PROGRA | xxxxxxx | EYKJAV | MPLETE | HUMANI | PROSPE | VERYTH | xxxxxx |
| TUNATE | xxxxxxx | SHINGT | UNATEL | NUCLEA | RTUNAT | CLIMAT | FRIEND | PRINCI | OSPHER | ITHDRA | GLIMPS |

6-Gram Feature Map of Query in Neural Filter

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| ENDSHIP | xxxxxxx | xxxxxxx | xxxxxxx | xxxxxxx | PRINCIP | xxxxxxx | ERGETIC | POLITIC | GREATES | EQUALIT | OMPLETE |
| AINTAIN | RESTRIC | ENEFICI | xxxxxxx | xxxxxxx | PROSPER | HINGTON | xxxxxxx | MAINTAI | PARTIES | PERMITT | SUCCESS |
| xxxxxxx | xxxxxxx | ADMINIS | RDINARY | CONNECT | POWERFU | YKJAVIK | xxxxxxx | xxxxxxx | xxxxxxx | AORDINA | NTEREST |
| xxxxxxx | xxxxxxx | BVIOUSY | REDUCIN | PROGRAM | xxxxxxx | EVERYTH | NTIBALL | STRATEG | UTUALLY | xxxxxxx | NECESSA |
| RCUMSTA | xxxxxxx | xxxxxxx | xxxxxxx | QUALITY | xxxxxxx | DYNAMIS | NRESTRI | NDEPLOY | xxxxxxx | ISAPPOI | ITHDRAW |
| xxxxxxx | FRIENDS | xxxxxxx | xxxxxxx | EREOTYP | REYKJAV | ECESSAR | TUNATEL | TANDOFF | TASTROP | xxxxxxx | xxxxxxx |
| IRCUMST | xxxxxxx | COMPROM | xxxxxxx | YNAMISM | PEACEFU | LATIONS | xxxxxxx | xxxxxxx | ECURITY | ESEARCH | xxxxxxx |
| FLEXIBI | TRUMENT | CONTOUR | xxxxxxx | AMERICA | xxxxxxx | TMOSPHE | IENDSHI | ASTROPH | xxxxxxx | DISPLAY | DISAPPO |
| NUCLEAR | ROSPERI | EACEFUL | xxxxxxx | xxxxxxx | APPOINT | COMPLET | CUMSTAN | ATANTLY | ATMOSPH | xxxxxxx | MERICAN |
| NTIRELY | EEMENTS | xxxxxxx | RIENDSH | LLISTIC | ANYTHIN | OWERFUL | xxxxxxx | ATASTRO | UMSTANC | APPROAC | PPOINTM |
| SOCIALI | CLIMATE | xxxxxxx | UCCESSF | TLESSNE | SECURIT | OSPERIT | OVEMBER | xxxxxxx | EYKJAVI | xxxxxxx | AGREEME |
| xxxxxxx | xxxxxxx | xxxxxxx | xxxxxxx | xxxxxxx | STEREOT | SUPPORT | STROPHE | ORDINAR | xxxxxxx | SHINGTO | PPROACH |

7-Gram Feature Map of Query in Neural Filter

In all the above simulations, the best size for *n* seemed to be the average word length of the language (*best* means the most efficient trade off between computational efforts and retrieval or representation quality).

*Results for the Neural Filter Algorithm Based on N-Grams with Spaces*

The n-grams as shown above are all without the incorporation of spaces. In the next simulations the spaces were used too, to gain a better insight in the contextual relations between words. Without spaces one actually only determines keyword parts. By incorporating the spaces, relations between words are taken into account as well. Without these relations, n-grams are nothing more than a keyword search method without a dictionary of predetermined keywords. The spaces are represented by *[*. This is due to the fact that this is the next ASCII character after the *Z*, which facilitates programming.

| | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| xxx | [EV | [EQ | WAS | CAT | BER | RYT | WID | SYS | [NE | [MA | TIC | WES | T[W | K[C | L[R | INK | INC | MOS | OOD |
| DAN | [AT | [AB | LET | xxx | SEC | FAC | UES | FIG | RIC | HER | TER | TES | xxx | POW | PPR | GOT | END | INT | OIC |
| ONS | NEG | OAC | NES | OUR | xxx | EAT | NAT | MED | IEV | GUR | VER | RAW | FAV | xxx | xxx | AND | UNR | ENT | NIT |
| OOP | N[G | N[S | LIS | NEC | ACT | UCT | IET | URT | FUS | UAR | EAR | EQU | EGI | IMA | EFU | A[W | ANG | ANS | FIR |
| I[T | IPP | Y[A | H[W | RST | EXT | ITS | NST | UCC | ARD | ERF | EAS | EVE | M[A | ELA | ACE | CIA | QUA | RMA | VIE |
| NGT | Y[G | F[D | LOV | GOV | ARC | NDS | [ST | ECT | ESS | ERS | EAC | xxx | I[E | MLY | xxx | CEF | RMI | ANI | ENE |
| L[T | G[S | T[R | D[F | xxx | ORC | OSS | NTO | [SH | IRC | YTH | xxx | ABO | AKN | AIN | EEM | REY | RML | UNF | UNA |
| GET | [IT | C[S | GGR | DEF | CUR | [CL | [EL | [SO | [FO | GGL | EGO | ATO | ISP | EYK | MEN | MAI | FUL | IMM | EMA |
| [EX | LIC | DIS | DUC | CES | CIP | TRO | GTO | OSP | [DO | xxx | UPP | EXP | EAP | UIP | MIN | YKJ | ION | MOM | IOU |
| LUD | [US | RUC | HAL | TEL | TMO | BAL | PRO | OBO | [CO | V[P | ELO | ALK | ACH | IZO | Y[N | YON | N[N | NOU | MPS |
| PUL | ERO | REL | REO | TYP | RYO | GEN | LAY | ORM | [AM | xxx | CHO | S[H | UCL | UGG | ILL | [PO | OPO | ROP | OOR |
| NFO | IAL | EAK | FEN | RAN | HIN | LIM | LIZ | [IN | [UN | [ON | SHO | C[E | AGG | RCO | DOO | OMP | L[L | LOP | D[P |
| N[P | IGH | xxx | xxx | CUM | RIN | PEN | HUM | [MI | [MO | TON | T[N | DWI | THI | C[N | CON | RNM | H[N | SPH | SOV |
| xxx | RTH | SSR | SSU | ONN | GAI | DMI | GRE | xxx | SOC | T[I | XTE | TSE | RTU | SHI | V[I | R[N | R[U | DOF | ROW |
| xxx | DSH | HDR | [GU | HOI | [FI | [RE | FRI | FRU | SWE | SAR | TRA | TRI | TEN | CCE | ALI | R[I | T[U | TOU | BOD |
| [TR | [DE | [HA | [LI | L[U | [SU | [WE | ORA | [GA | C[C | SSF | TRU | RFU | xxx | SSA | SPE | XPE | T[A | D[G | TOT |
| NCE | OCI | NTE | KJA | PEA | LAB | [QU | [WA | A[B | ALT | xxx | ARY | AVI | REM | QUI | R[E | C[A | T[F | THD | T[D |
| OTI | NGE | OKE | NNE | OYM | PER | NER | GUA | Y[R | xxx | ARM | ISM | URI | ERN | ISA | RSE | HIE | S[G | R[T | R[S |
| LLY | PLE | [LA | PME | OVI | OFF | OGR | [FR | ELS | Y[D | AGR | ITA | ATM | USY | ICY | GHE | GLI | xxx | R[R | U[F |
| OPM | OPE | [PE | M[F | PRI | PSE | OWE | [PR | FOR | BOR | ECE | ATI | USA | ITY | ETE | FLE | CLE | F[R | ROA | R[A |

3-Gram with Spaces Feature Map of Query in Neural Filter

As in the earlier simulations, the window size can be increased easily to 7-grams (or higher). Below a map for the 7-gram with spaces and the 8-grams with spaces are given.

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| xxxxxxx | RFUL[US | [MUTUAL | [WEAKNE | xxxxxxx | TRU[HOP | SHOW[ST | UARD[GA | [NECESS | [IMPLEM | [HALT[N | xxxxxxx |
| [FORC[E | ZON[GLI | [GAIN[S | RN[CLIM | TRONG[S | RICAN[W | xxxxxxx | [CONNEC | xxxxxxx | TUNATEL | UL[USSR | [TALK[C |
| UCT[ELI | UL[LABO | xxxxxxx | TY[GENE | ROV[INT | WITHDRA | RANG[PE | TEND[FR | UIR[APP | [EXTEND | RM[FORC | S[CONTO |
| WID[RAN | [INTERN | [PROSPE | [SOCIAL | [ARM[FO | xxxxxxx | ROSPERI | [UNFORT | xxxxxxx | URITY[D | T[NUCLE | XTEND[F |
| RTUNATE | xxxxxxx | [STANDO | ST[SHOW | xxxxxxx | UR[MUTU | [AGREEM | xxxxxxx | REACT[R | TRATEGI | RATEGIC | R[EXTEN |
| [APPROA | UALLY[A | xxxxxxx | TH[NECE | xxxxxxx | xxxxxxx | [SECURI | xxxxxxx | [POWERF | [IMPROV | [USSR[A | xxxxxxx |
| xxxxxxx | [CIRCUM | VOR[IMM | T[SDI[P | SSR[ARM | [DEFEND | SECURIT | xxxxxxx | TABILIT | xxxxxxx | SARY[GU | TALK[CO |
| SSARY[G | Y[ITSEL | VERNMEN | [ELS[CL | UCING[S | YTH[NEC | [GLIMPS | VIK[MOM | TYP[CON | [NUCLEA | RTICL[T | TEST[RE |
| Y[CATAS | R[APPRO | [REMAIN | RPOS[AC | STRUGGL | USSR[AR | ROPOS[R | RY[GUAR | SHIP[CO | [STRONG | RTY[TRE | RMLY[DE |
| xxxxxxx | xxxxxxx | TEST[PO | TICL[TR | STROPHE | RYTH[NE | T[FIRML | [DOOR[R | SPERITY | [FIRMLY | R[REMAI | RD[GAIN |
| [TREATY | xxxxxxx | UNFORTU | RESEARC | [PERMIT | RIENDSH | [EVERYO | xxxxxxx | U[HOP[N | T[DEPEN | [DESTRU | [REYKJA |
| TMOSPHE | SOCIALI | [NEGOTI | T[INTER | [PRINCI | [PEACEF | T[DESTR | SS[PEAC | [UN[SUP | TTED[LA | Y[GUARD | xxxxxxx |

7-Gram with Spaces Feature Map of Query in Neural Filter

| | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| xxxxxxxxx | S[PEACEF | AR[STAND | ESS[COOP | N[REPRES | D[DANGER | T[DEPEND | H[WID[RA | ITY[GENE | EQUIR[AP | xxxxxxxxx | C[EQUIPP | D[GAIN[S | Y[AMERIC | AVIK[POL |
| ERICAN[W | EST[POSS | CT[WEAKN | [UN[SUPP | R[MUTUAL | ITTED[LA | STR[PURS | TRAORDIN | E[UNFORT | ATEST[PO | [VIEW[MU | QUIR[APP | K[MOMENT | xxxxxxxxx | LY[MATTE |
| GTHEN[PR | I[TALK[C | N[SUPPOR | M[FORC[E | TURNAROU | S[GOODWI | IR[APPRO | T[NUCLEA | TYP[CONS | N[PERMAN | ARD[GAIN | NUIN[TAL | R[ARM[FO | xxxxxxxxx | M[TREATY |
| [REACT[R | A[BALL[W | EQUALITY | DWILL[PR | [UNFORTU | xxxxxxxxx | NTIBALLI | xxxxxxxxx | IT[ST[CI | L[PRINCI | LS[CLEAR | PROSPERI | F[DOUBLY | G[STRATE | xxxxxxxxx |
| L[PROPOS | ISTR[PUR | T[DESTRU | RY[CATAS | xxxxxxxxx | [UN[FORT | SSARY[GU | RT[MAINT | ERN[POLI | C[PROSPE | [WOUL[RE | ER[MILIT | xxxxxxxxx | L[USSR[A | R[IMMEDI |
| I[IMPROV | ORC[EQUI | D[RANG[P | xxxxxxxxx | [UNDERST | DUR[IMPL | xxxxxxxxx | OR[IMMED | PROPOS[R | PROV[INT | xxxxxxxxx | OR[REMAI | N[SOCIAL | xxxxxxxxx | ORTUNATE |
| IRMLY[DE | N[EROD[U | [STRUGGL | W[STRONG | LY[DECID | L[EQUALI | [WID[RAN | PS[CONTO | N[PROV[A | STIPUL[R | IT[ANYTH | xxxxxxxxx | PURSU[CO | F[REQUIR | RYTH[NEC |
| NT[RADIC | ARTY[TRE | EST[REDU | EYKJAVIK | FUL[USSR | CT[NUCLE | DSHIP[CO | CUMSTANC | xxxxxxxxx | EXTEND[F | RU[HOP[N | xxxxxxxxx | [STRATEG | OSS[GOOD | ARTICL[T |
| R[EXTEND | T[FIRMLY | ASON[DEN | AR[EVERY | DUCING[S | ARY[GUAR | ORIZON[G | STRONG[S | ATORY[RE | ET[FIRML | xxxxxxxxx | Y[DEFEND | [STEREOT | xxxxxxxxx | STRUCT[N |
| TUALLY[A | CURITY[D | ESSARY[G | CT[ELIMI | AVIK[MOM | xxxxxxxxx | [WEAKNES | OVIET[FI | xxxxxxxxx | xxxxxxxxx | Y[MUTU[F | xxxxxxxxx | RY[GUARD | xxxxxxxxx | R[IMPLEM |
| GUARD[GA | EST[SHOW | xxxxxxxxx | ER[EXTEN | xxxxxxxxx | xxxxxxxxx | N[GLIMPS | ASTROPHE | MUTU[PRE | ATIONS[H | XTEND[FR | ERMITTED | [SHOW[ST | R[CONDIT | xxxxxxxxx |
| xxxxxxxxx | ARM[FORC | xxxxxxxxx | NT[SDI[P | EVELOPME | R[APPROA | ERYTH[NE | [USSR[AR | ERFUL[US | xxxxxxxxx | ATMOSPHE | Y[NOVEMB | UTU[PREC | ESS[PEAC | xxxxxxxxx |
| L[WASHIN | xxxxxxxxx | [WITHDRA | xxxxxxxxx | xxxxxxxxx | xxxxxxxxx | Y[ITSELF | T[ST[CIR | ITIC[SOV | I[ELS[CL | FRE[UNFO | TRUGGL[P | USSR[ARM | PRINCIPL | RTUNATEL |
| K[POLICY | OSPERITY | ET[DESTR | ATEGIC[N | TRONG[ST | C[AGREEM | ITY[AMER | DRAW[ABM | ITY[DOOR | NTERN[PO | [SUPPORT | HUMANITY | DUCT[ELI | OVERNMEN | FRIENDSH |
| ISARM[AC | xxxxxxxxx | [SECURIT | xxxxxxxxx | xxxxxxxxx | SS[PEACE | YTH[NECE | [SOCIALI | LY[DEFEN | R[STANDO | TRATEGIC | xxxxxxxxx | xxxxxxxxx | M[ACHIEV | T[ELIMIN |

8-Gram with Spaces Feature Map of Query in Neural Filter

*Results of Neural Filter Algorithm based on a Markov Chain over Keywords and N-Grams* [7]

Now, an $n^{th}$ order Markov chain over both words or large n-grams is presented. First, the learning text is preprocessed to determine the words or n-gram present. As long as the system is not *out of memory*, the model stores words or n-grams and assigns random codes to them.

Next, combinations of three words are taught to the neural net. Suppose a statistical method: then one needs either (number of words)$^3$ memory elements or sophisticated count, order, normalization, generalization and association methods [Brown et al., 1990], [Jelinek, 1989 & 1991a-c]. Here, the same amount of memory is used at the same learning speed as in the character trigram method. The map shown below is a small part of a large 15 x 15 feature map.

| | | |
|---|---|---|
| INTERESTS-DISARM-PARTY | IMPROV-INTERN-CLIMATE | ATMOSPHER-NUCLEAR-STANDOFF |
| FAVOR-OUR-TALK | DON-ARM-RAC | AGREEMENT-ATMOSPHER-NUCLEAR |
| DISARM-ACHIEV-PEAC | WEAPON-DEVELOPMENT-DENOUNC | TRU-DYNAMISM-TALK |
| DENOUNC-REL-PRINCIPL | INTERN-POLITIC-SOVIET | AGREEMENT-NONDEPLOYMENT-SPAC |
| FORTH-WID-RANG-EQUIPP | EVERYTH-NECESSARY | PARTY-TREATY-ENTIRELY |
| ENTIRELY-INTERN-POLITIC | STRENGTHEN-PROV-AGREEMENT | ITSELF-DOUBLY-IMPOSS |
| PEAC-PROSPERITY-OBVIOUSY | SUPPORT-MAINTAIN-ABM | REDUCT-COMPLET-DESTRUCT |
| ARM-FORC-EQUIPP | TRU-HOP-N | WEST-SHOW-STRONG |

Upper Left Part (3x8) of Tri-Words Feature Map of Query in Neural Filter

To derive this map, the learning text was preprocessed so all possible non-relevant words were determined. Two hundred eighty one words were found in the text on the *Nuclear Weapons Restriction Talks*. A list of these words is given on the next page.

[7] The results for the markov chain over keywords were about the same as the ones over large n-grams. The only difference was in the need to define a dictionary in advance. Therefore these two algorithms are described in the same paragraph.

| | | | | |
|---|---|---|---|---|
| AGREEMENTS | TOWARD | PARTIES | GRIGOROVICH | FLEXIBILITY |
| ELIMIN | COLLECT | PROVOC | ENERGETICALLY | ERECT |
| CHOREOGRAPHER | UNIT | NOBODY | CONVENT | EXTRAORDINARY |
| GROW | AGAIN | MED | AREA | EXPENS |
| AGRE | ATTENT | KUKHAR | TRY | ERA |
| SOVIET | ANSWER | HERO | TALK | DISCUSS |
| REG | VIT | HARBOR | TODAY | AGRICULTUR |
| STRATEGIC | UNDERMIN | DETERMIN | SDI | ACQUIR |
| SERI | RESPOND | COEXIST | REACHING | WITHDRAW |
| CONSIDER | COURSE | THOUGH | PROSPERITY | WEAPON |
| SYSTEM | AGGRAV | PRESIDIUM | PERMITTED | TEST |
| AGREEMENT | MOSCOW | PURSU | POSIT | REDUCT |
| ANYTHING | ENTIRELY | PRECLUD | ORDER | PROVINCE |
| BROKEN | UNRESTRICT | NIKOLAEVICH | NEGOTI | PROGRESS |
| ART | STRENGTHEN | INDICATORS | DEVELOPMENT | PREC |
| STABILITY | REDUCING | FACT | DELEG | LIMITA |
| MUTU | KREMLIN | DEPEND | BALL | IVAN |
| HOV | JANUARY | COURT | ACHIEV | INTERESTS |
| ACT | ON | CONNECT | ADD | INERTNESS |
| DYNAMISM | ITSELF | THREATEN | TITL | IOFF |
| CONDIT | CIRCUMSTANC | PROV | SOCIALIST | FULFILL |
| SUPPORT | ANTIBALLISTIC | PERMANENT | OBVIOUSY | EROD |
| COMRAD | DELIBERATELY | POSS | MUTUALLY | EDIT |
| FRAGIL | SUPREM | PROBLEM | BEHALF | CONCLUS |
| SUR | PROGRAM | POSSIBILITY | ARSEN | WISH |
| CHOREOGRAPHIC | LEARN | OUTSTAND | ABM | WOUL |
| SICKL | DECEMBER | NUCLEAR | ADMINISTR | UPON |
| SURV | WASHINGTON | LESS | DESERV | USSR |
| TREATY | DISC | IMPLEMENT | BAS | UNIVERS |
| SALT | REGRET | HEROISM | TOT | TURNAROUND |
| MENTESHASHVILI | REPRESENT | GOVERNMENT | SECRETARY | PRODUCT |
| SIGN | PURPOS | DISPLAYED | QUEST | PREVENT |
| FARM | IVANOVICH | DISPLAY | IMPOSS | KINGSBURY |
| SPAC | DEC | CHAIRMAN | DISAPPOINTMENT | INDEFINITELY |
| ST | CLOS | CHOIC | CAS | HAMMER |
| BECOM | BEGINN | COMPROM | BROKE | HIGHER |
| TRU | ACADEMIC | CONDUCT | USA | GROMYKO |
| EXPLOIT | DECRE | BRONZ | THEATER | AR |
| SECURITY | CAUS | BREED | RAC | WANT |
| INTEREST | REDUC | BENEFICI | PEAC | WHETHER |
| POLICY | PAC | BLATANTLY | OVERCOM | TARGET |
| VIEW | MATTER | AWRD | LABOR | POLITIC |
| DON | LIVESTOCK | STIPUL | DENOUNC | NONDEPLOYMENT |
| UPSET | LIMIT | RESEARCH | BIRTHDAY | MAINTAIN |
| MAK | COMMITT | PRESENT | ARM | LABORATORY |
| DUR | SERVIC | HONEST | SAL | ILYICH |
| BUST | OUR | DEVELOP | REFUS | ISSU |
| INTERDEPENDENC | FINALLY | BOLSHOI | RADIC | INPUT |
| DOUBLY | FAVOR | WITHIN | REPEAT | HOSTILITY |
| DEPR | FRUITLESSNESS | REYKJAVIK | PROPOS | GOLD |
| DEFENS | DECID | JOURNALIST | PERIOD | GENUIN |
| REASON | DETON | GENEVA | OTHER | DESPIT |
| INCLUD | CONTINU | PLANET | MONTH | AWARD |
| GORBACHOV | ARTICL | NOVEMBER | LENIN | |
| BIRTHPLAC | AMERICAN | LEADERSHIP | INSTRUMENT | |
| UN | WAR | FAST | HONOR | |
| COOPER | TRANSLATOR | PARTY | HOP | |

Words Used in 3-Word Neural Filter.

Then, the text was processed again and trigrams on words were determined and fed into the feature map. After a number of passes, the map holds a proper representation of the learning text. The values stored in the neural weights are summarized on the next page. Elements are represented from the upper left corner of the map down to the under right one.

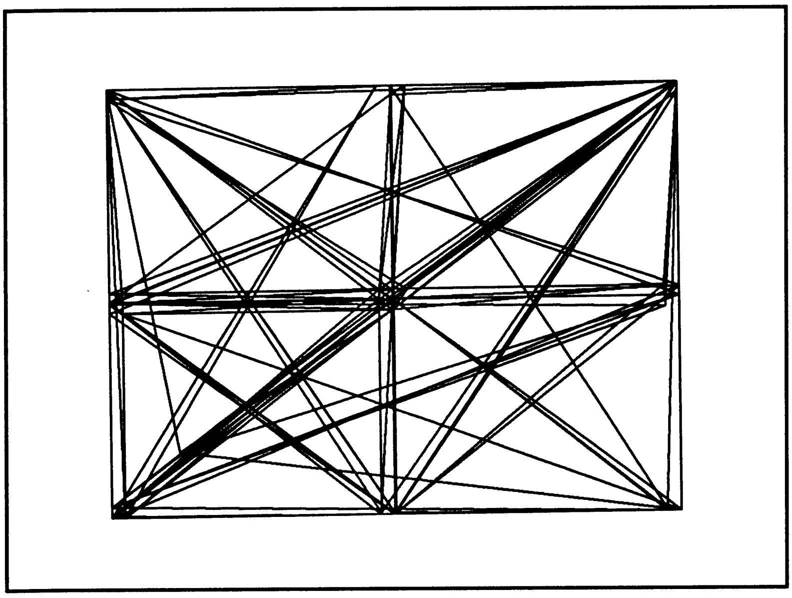
AGREEMENT-WITHIN-ANTIBALLISTIC
PRECLUD-POSSIBILITY-SECURITY
THREATEN-HIGHER-INTERESTS
AGRE-REDUCING-STRATEGIC
AMERICAN-GOVERNMENT-DEPEND
AGREEMENT-NONDEPLOYMENT-SPAC
PEAC-FRAGIL-PLANET
DEFENS-SYSTEM-DUR
GENEVA-TALK-SOVIET
SOVIET-UN-SUPPORT
HOP-REDUC-NUCLEAR
ANYTHING-DON-EROD
UN-SUPPORT-MAINTAIN
TREATY-USA-CONSIDER
FACT-ARTICL-TREATY
NUCLEAR-CONVENT-WASHINGTON
ITSELF-DOUBLY-IMPOSS
USA-CONSIDER-BENEFICI
DEPEND-CHOIC-INCLUD
PROVOC-ACT-BROKE
DEFENS-ISSU-AGREEMENT
PREC-UNIVERS-WHETHER
WITHDRAW-ABM-TREATY
ADMINISTR-PURSU-COURSE
PURSU-COURSE-MAK
USA-WITHDRAW-ABM
NUCLEAR-ARM-AREA
SYSTEM-DUR-MUTUALLY
MAINTAIN-ABM-TREATY
ISSU-CAUS-DISAPPOINTMENT
MUTU-PREC-UNIVERS
CHOIC-INCLUD-PEAC
WASHINGTON-QUEST-CAS
OVERCOM-ST-FRUITLESSNESS
INSTRUMENT-TREATY-ITSELF
PROGRAM-SOVIET-UN
AMERICAN-POSIT-ADMINISTR
COEXIST-PEAC-FRAGIL
ARM-NUCLEAR-CONVENT
SOVIET-INSTRUMENT-TREATY
SOVIET-AMERICAN-REPRESENT
PARTY-TREATY-ENTIRELY
MAINTAIN-INDEFINITELY-MATTER
INDEFINITELY-MATTER-FACT
WASHINGTON-COURT-OUR-
PLANET-QUEST-SUPPORT
MUTU-FLEXIBILITY-POSS
SUPPORT-TALK-OVERCOM
ABM-TREATY-FAVOR
AGREEMENT-SALT-CONDUCT
AMERICAN-WISH-PEAC
PERIOD-ANSWER-CONDIT
BAS-DEFENS-SYSTEM
STABILITY-FAVOR-MAINTAIN
STIPUL-REASON-DENOUNC
DON-EROD-UNDERMIN
NUCLEAR-WEAPON-ERA
ARM-PREVENT-ARM
ATTENT-ST-AGAIN
RAC-REDUCT-ELIMIN-
WANT-UNRESTRICT-ARM
AGREEMENT-USA-DETERMIN
PROPOS-SOVIET-INSTRUMENT
TEST-REFUS-DISCUSS
EXTRAORDINARY-CIRCUMSTANC-THREATEN
GOVERNMENT-DEPEND-CHOIC
REDUCING-STRATEGIC-NUCLEAR

ACT-BROKE-STRATEGIC
REYKJAVIK-TRY-ACHIEV
COURT-OUR-FINALLY
ARM-LIMITA-AGREEMENT
ARM-AREA-BALL
PROGRESS-ISSU-LIMIT
OTHER-REPEAT-SURV-
REASON-DENOUNC-EXTRAORDINARY
HIGHER-INTERESTS-PARTY
ACHIEV-PROGRESS-ISSU
RESEARCH-ABM-AREA
ACHIEV-ENERGETICALLY-SUR
FINALLY-DECID-WANT
ACHIEV-PEAC-AGREEMENT
CAS-PARTIES-DISPLAY
POSS-REACHING-COMPROM
LIMIT-PERMITTED-LABORATORY
STRENGTHEN-PROV-AGREEMENT
AREA-BALL-WASHINGTON-
LIMIT-REDUC-ARM
USA-DETERMIN-LIMIT
UNDERMIN-ABM-TREATY
DOUBLY-IMPOSS-AGRE
ANTIBALLISTIC-DEFENS-ISSU
DEC-AMERICAN-GOVERNMENT
DENOUNC-EXTRAORDINARY-CIRCUMSTANC
PERMITTED-LABORATORY-RESEARCH
WISH-PEAC-PROSPERITY
TREATY-STIPUL-REASON
TURNAROUND-TALK-WOUL
PROSPERITY-AMERICAN-WISH
FAVOR-STRENGTHEN-PROV
DUR-MUTUALLY-AGRE
DESERV-CLOS-ATTENT-
FRUITLESSNESS-INERTNESS-ACQUIR
DISAPPOINTMENT-REYKJAVIK-AMERICAN
POLITIC-INTERDEPENDENC-PRECLUD
SUR-RADIC-TURNAROUND
AREA-PROPOS-SOVIET
NOBODY-DEC-AMERICAN
PEAC-PROSPERITY-OBVIOUSY
SUPPORT-MAINTAIN-ABM
PARTIES-DISPLAY-MUTU
COMMITT-ITSELF-CONDUCT
LESS-COOPER-DESPIT
ITSELF-CONDUCT-NEGOTI
QUEST-CAS-PARTIES
CONTINU-TEST-REFUS
EXPENS-OTHER-REPEAT
DETERMIN-LIMIT-PERMITTED
POLITIC-SOVIET-UN
REACHING-COMPROM-AGREEMENT
LABORATORY-RESEARCH-ABM
DEPR-HOP-REDUC
AR-REGRET-ADMINISTR
STRATEGIC-NUCLEAR-ARM
DECID-WANT-UNRESTRICT
GENEVA-BECOM-LESS
MATTER-FACT-ARTICL
EROD-UNDERMIN-ABM-
ISSU-AGREEMENT-NONDEPLOYMENT
REDUCT-ELIMIN-ARM
CIRCUMSTANC-THREATEN-HIGHER
OUR-FINALLY-DECID
TREATY-ITSELF-DOUBLY
ELIMIN-ARM-NOBODY

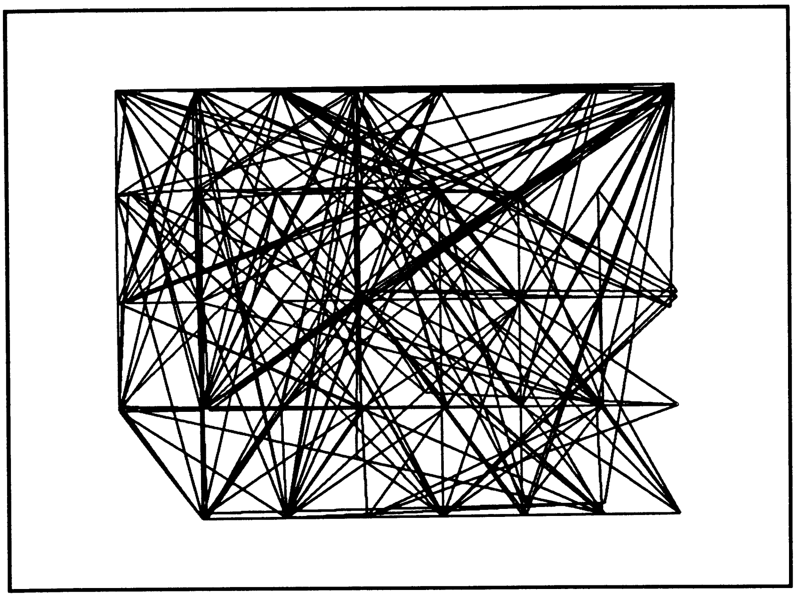
3-Word Elements Represented by a 15x15 Map (Summarized for reasons of space)

An interesting question concerns the convergence of the map toward a proper representation of the text. Therefore, two methods have been used. First the values of the sensors are graphed in such a way, the self-organizing character becomes clear. Second, the evolution of the error as function of the learning cycles was monitored during the learning process.

Two sensor values, $\xi_1$ and $\xi_2$ can be drawn in an (x,y) graph, where x is represented by the value of $\xi_1$ and y by $\xi_2$. By interconnecting neighbouring neurons, self- organization is represented by a perfect rectangular figure if the distribution of the input values is comparable to the size and form of the feature map. If one works with artificial codes (not natural, but assigned from look up tables) and the number of used codes differs from the dimension of the map, regular, but different figures develop. The pictures below represent two sensors from the first trigram simulation and the last word simulation. In the trigram simulation (left), the codes used were (0.0, 0.5, 1.0). This map is well organized. In the word simulation (right) 10 different sensory values were used. But, only a certain part was used by the word coding procedure ($9^{10}$ codes were available, only $250^3$ were used). Therefore the graph is slightly smaller on one sight (not used codes). Here too, one can observe a regular structure (albeit less then in the trigram case), indicating some form of self-organization.



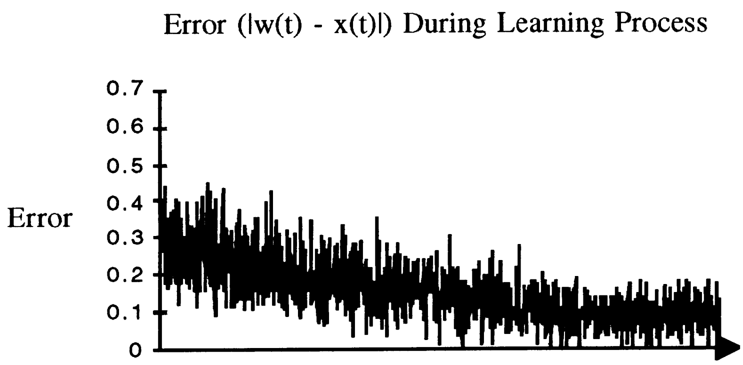
Sensor Value Graph Trigrams ($\xi_1$, $\xi_2$)



Sensor Value Graph Words ($\xi_1$, $\xi_2$)

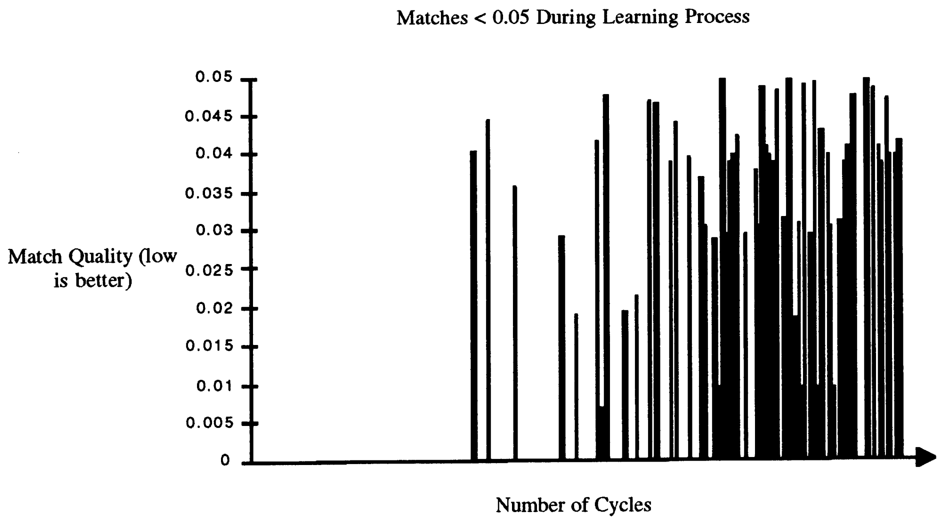
Please notice, these graphs represent only 2 sensors of 9. All other graphs looked similar.[8]

By measuring the error during the learning process: $\|w_r(t) - x(t)\|$, an insight in the convergence properties of the neural net can be obtained. First, one has to understand that this neural net is used as a selection- and ordering device. Due to a smaller size than needed, only the most frequent n-grams are remembered (or learned properly), all others are forgotten, or overruled. Therefore, the average error will remain high (due to non frequent n-grams). In the first graph the total error in time is plotted (see next page). The global character of the graph is decreasing. The high errors on the right are non-frequent trigrams that *must* be forgotten (these are errors of n-grams which are continuously being bounced out).

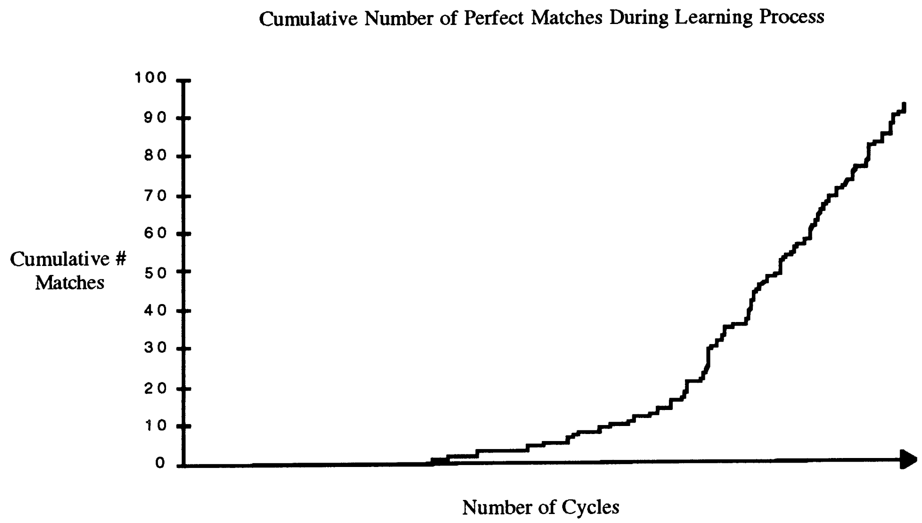



[8] These graphs clearly indicate that the distribution of the n-gram and keyword clusters is completely different from the topology of the feature map. Therefore it is very difficult to interpreted the *topological map* of language. Future research concentrates on other types of feature maps, which must be able to represent such distributions better. In this report we try to use a two-dimensional, rectangular, and homogeneous feature map for the representation of a (probably) non-dimensional, randomly connected, and clustered input: language.

The next graph plots the error if it is smaller than 0.05. By plotting these bars, one sees that the frequency of perfect hits increases in time: the density of bars is much higher at the right side of the graph than at the left. This indicates that the model is getting better at representing n-grams. The number of small errors increases as the learning continues.




The following graph represents the *cumulative* number of perfect hits (> threshold) in time. In the beginning, there are no hits at all. At a certain time, the number increases exponentially (self-organization starts). At the end, the number of hits stays constant (resulting in a linear increase of the cumulative value). These three graphs indicate that the net does indeed learn certain n-grams and it gets better at this task, the longer it learns, up to the moment the maximal capacity of the map is reached. From then on, only the most frequent ones will be learned.

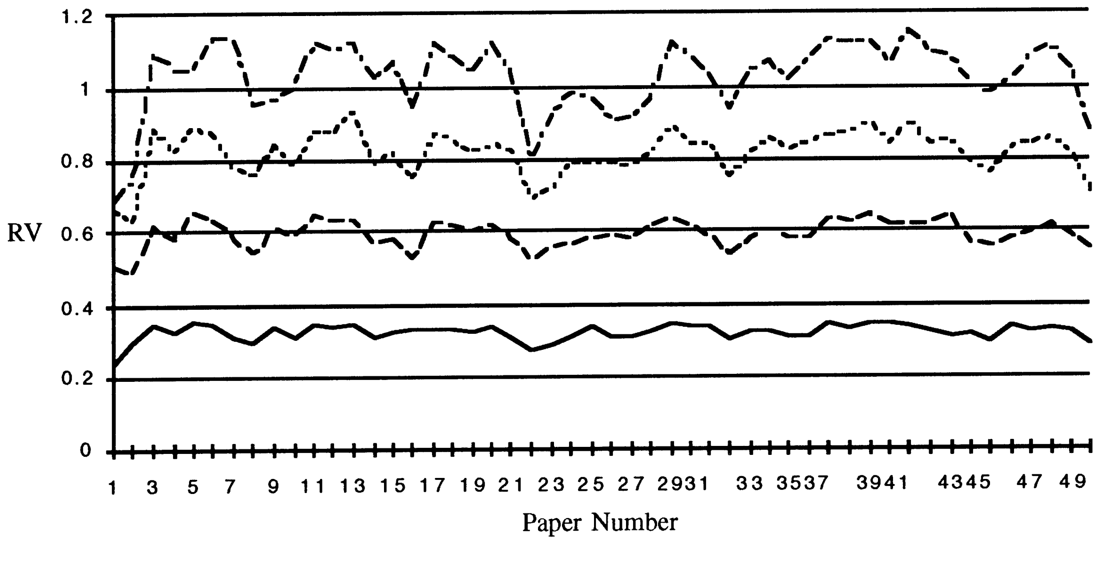


## Retrieval Results of the Neural Filter

### *Retrieval Results of the Neural Filter Based on N-Grams*

The selection quality is a measure that cannot be given freely without being impartial. During the simulations, many reasonable decisions were made by the selection algorithm. Much depends on the threshold, but the selections were as least as good as the statistical ones. Still, the neural ones where much easier to implement, where the statistical ones required much programming. The generalizations made were very interesting, although, it must be admitted, this was not implemented in the statistical algorithm (due to the large amount of efforts that it would have taken).
On the next page, an overview of the results for the 4-, 5- and 6-grams is given. Per paper, a short description of the contents can be found. On the right hand, retrieval values are given. The smaller the values, the higher the correspondence.
The retrieval phase uses several different functions. The proposed function (average error per n-gram in the retrieval phase) separates related text parts clearly from non-related. Yet, the differences are quit small. That's what makes the Pravda interesting as a corpus[9]. There is much noise from words like *comrade, socialism, hero,* etc., making the retrieval phase more difficult (these words were not eliminated, but should have been). By counting the average error per n-gram as well as the number of perfect hits, a better discrimination function is found. Generalizations caused by both the n-gram formalism and the Kohonen feature maps could be observed during the retrieval phase.
The retrieval values (see following page) can be plotted in a graph. The lines represent (from lower to upper part of graph) the retrieval values for 50 text parts of the Pravda of the 4-gram, 5-gram, 6-gram and 8-gram analyses. Low values indicate low errors and thus high correlation. The first article is the same as the learning text (because not all n-grams are taught to the neural map, a number of errors remains) It might be clear that the separation becomes better as the window size gets larger. This graph is based on the first selection rule: the normalized total error per text part. This is in fact a very negative approach.

Retrieval Values (RV) Information Filter (4-, 5-, 6-, and 7-grams)




[9] Others say the Pravda is a very bad corpus for information retrieval research because the language used involves syntax without semantics: the articles mean nothing, its just propaganda. however, we prefer the other view stating that the Pravda has a lot of noise, making it boring to read and therefore a useful application of IR techniques.

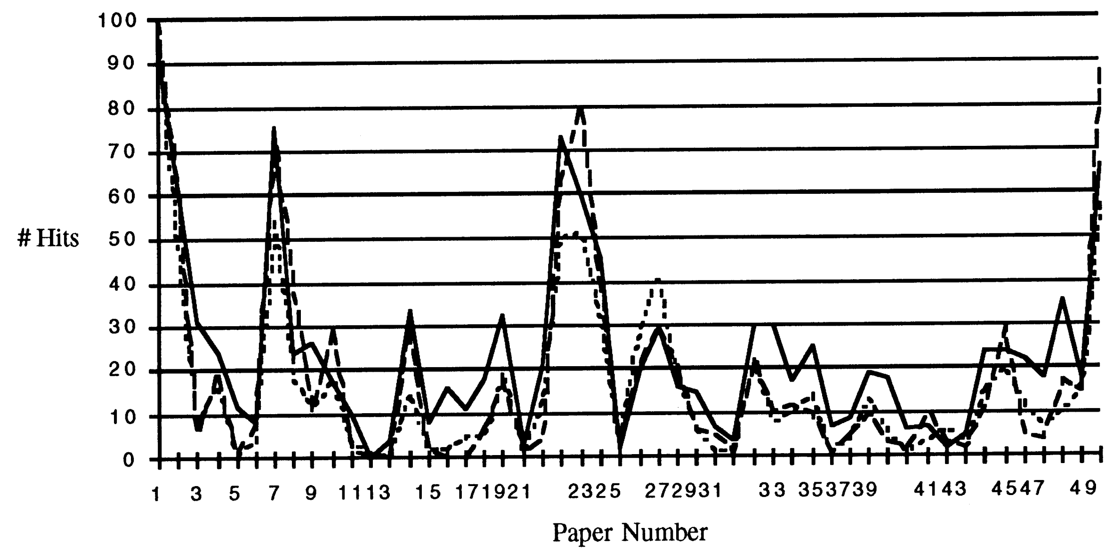
| # | Subject | Retrieval Values on Nuclear Query | | | |
|---|---|---|---|---|---|
| | | 4-gram | 5-gram | 6-gram | 7-gram |
| 0* | Weapons talk & socialist medals | 0.237318 | 0.509591 | 0.662473 | 0.689539 |
| 1 | New years wish of M. Gorbachov | 0.289728 | 0.488024 | 0.632248 | 0.744335 |
| 2 | A day in the life of a museum director | 0.341379 | 0.617132 | 0.884395 | 1.097080 |
| 3 | Soviet Literature | 0.322802 | 0.585385 | 0.823094 | 1.052009 |
| 4 | A day in the life of ... | 0.354099 | 0.657095 | 0.893633 | 1.052009 |
| 5 | A 2nd world war hero | 0.343971 | 0.640814 | 0.868803 | 1.139508 |
| 6* | Nuclear weapon talks | 0.313828 | 0.586899 | 0.777539 | 1.128579 |
| 7* | Peace demonstrations | 0.290589 | 0.546081 | 0.755270 | 0.951315 |
| 8 | New years wishes | 0.338634 | 0.613795 | 0.840996 | 0.961775 |
| 9 | World news | 0.310629 | 0.590616 | 0.782714 | 1.003833 |
| 10 | A fairy tale | 0.349794 | 0.647462 | 0.882928 | 1.122699 |
| 11 | On rabbits | 0.338259 | 0.639136 | 0.878360 | 1.101719 |
| 12 | Poem on carnival | 0.345716 | 0.638410 | 0.934767 | 1.126669 |
| 13 | Central comittee new years wishes | 0.314312 | 0.569742 | 0.787559 | 1.023176 |
| 14 | Labour news | 0.321815 | 0.586252 | 0.818649 | 1.075319 |
| 15 | On construction in the USSR | 0.332173 | 0.530753 | 0.748884 | 0.940438 |
| 16 | On transportation affairs | 0.333690 | 0.626034 | 0.869077 | 1.116766 |
| 17 | (Communist) party life | 0.334771 | 0.621255 | 0.855764 | 1.085567 |
| 18 | Space lift-off | 0.325455 | 0.599938 | 0.825772 | 1.046907 |
| 19 | Poem on nature | 0.340744 | 0.625286 | 0.841849 | 1.118814 |
| 20 | Story on a smoking teacher | 0.314774 | 0.588360 | 0.828015 | 1.041139 |
| 21* | USSR on foreign media | 0.273380 | 0.524630 | 0.692393 | 0.799374 |
| 22* | Nuclear weapons talk | 0.288171 | 0.559702 | 0.720814 | 0.926369 |
| 23 | On Afghanistan | 0.313803 | 0.573072 | 0.792759 | 0.986893 |
| 24 | On African countries | 0.336627 | 0.579585 | 0.790675 | 0.969926 |
| 25 | Economy: USA and EEC | 0.309720 | 0.593978 | 0.788340 | 0.908930 |
| 26 | Peoples Dreams | 0.311645 | 0.588900 | 0.783028 | 0.915919 |
| 27* | Satire on US Military | 0.322035 | 0.619409 | 0.815802 | 0.968615 |
| 28 | Story on Italy | 0.345008 | 0.637252 | 0.892476 | 1.120427 |
| 29 | Carnival | 0.337170 | 0.618419 | 0.838955 | 1.085120 |
| 30 | Driving a car in the USSR | 0.338137 | 0.588304 | 0.844606 | 1.032986 |
| 31 | The party's social policy | 0.302633 | 0.542491 | 0.750042 | 0.939680 |
| 32 | Economy news | 0.329482 | 0.587417 | 0.814469 | 1.045905 |
| 33 | Around the world news | 0.323436 | 0.605049 | 0.853771 | 1.068712 |
| 34 | Work circumstances | 0.312763 | 0.585953 | 0.830122 | 1.019806 |
| 35 | Automation in baking industry | 0.307923 | 0.583181 | 0.848865 | 1.076185 |
| 36 | Nature | 0.344264 | 0.634208 | 0.859321 | 1.130220 |
| 37 | The good old past | 0.331045 | 0.628162 | 0.877461 | 1.117062 |
| 38 | Theatre | 0.341568 | 0.649438 | 0.896636 | 1.123436 |
| 39 | Cambodia | 0.341709 | 0.617094 | 0.844380 | 1.064060 |
| 40 | Life in France | 0.337273 | 0.619542 | 0.895881 | 1.149020 |
| 41 | A letter to Santa | 0.328580 | 0.628589 | 0.845106 | 1.096221 |
| 42 | Sports | 0.312252 | 0.646526 | 0.852348 | 1.076769 |
| 43 | Industrial reports | 0.316586 | 0.568280 | 0.795445 | 1.011926 |
| 44* | Nuclear weapons talks | 0.293399 | 0.560232 | 0.762709 | 0.981953 |
| 45 | Gasoline | 0.339407 | 0.579052 | 0.828988 | 1.021601 |
| 46 | Product quality | 0.324115 | 0.596635 | 0.842714 | 1.082056 |
| 47 | On justice | 0.331254 | 0.615968 | 0.857014 | 1.112873 |
| 48 | On genetics | 0.327425 | 0.593382 | 0.815307 | 1.043186 |
| 49* | Nuclear Weapons | 0.286680 | 0.552238 | 0.706248 | 0.859771 |

Retrieval Values for Negative Selection (50 first papers from Pravda. #1 is the Learning Set)

* Somehow related to the query (checked manually for correlation in subject).

If we use the second retrieval function (a more positive one), the results are even better (see next graph). By counting the number of (almost) perfect hits and comparing the normalized value with a threshold (perfect retrieval is 100% in graph), the 7-gram learn text has a 90% retrieval value[10]. Even a small paragraph mentioning the subject resulting in an already high peak in the graph.

Number Perfect Hits Information Filter in Retrieval Phase (5-, 6- and 7-gram)




Not all papers of the Pravda have been scanned by hand. Even if this would have been done, it is really hard to express the amount of correlation in meaning. In many cases, it is quite easy to interpret the results in a different way[11]. Therefore, some related as well as some unrelated articles were inserted randomly in the test set. The model found them all with the proper retrieval values. Besides the inserted articles, all other articles found related to the query were in fact on the nuclear weapons talks between the USA and the USSR and *not* on conventional weapons, Chernobyl, other nuclear power plant, etc.

The determination of the most efficient retrieval function is a domain for study in itself. Obvious experiments can be done about combining a negative and positive learning rule. More mathematically based correlation functions can be incorporated, etc. This is a main topic of future research. Pointers can be found in the literature on statistical pattern recognition [Sammon, 1969], [Duda et al., 1973], [Small et al., 1974], [Fu, 1977], [Croft, 1977, 1980, 1981], [Bokhari, 1981], [Devijver et al., 1982], [Voorhees, 1985], [Siedlecki, 1988].

[10] This is quit high because the original learning text contains more words than the neural map can store. Therefore not all n-grams are remembered. Exactly these n-grams are responsible for the retrieval error. If we take a negative approach (the first retrieval function), this percentage will be much larger than in the case of a positive one. This is why the second retrieval function works better.

[11] A standard IR evaluation techniques compares the documents selected by hand with the documents selected by the computer. Here such experiments are not carried out due to the large amounts of time they consume, but future research does not exclude this method of evaluation. Moreover, a standard benchmark data collection would be very interesting. The author is not aware of the existance of one.

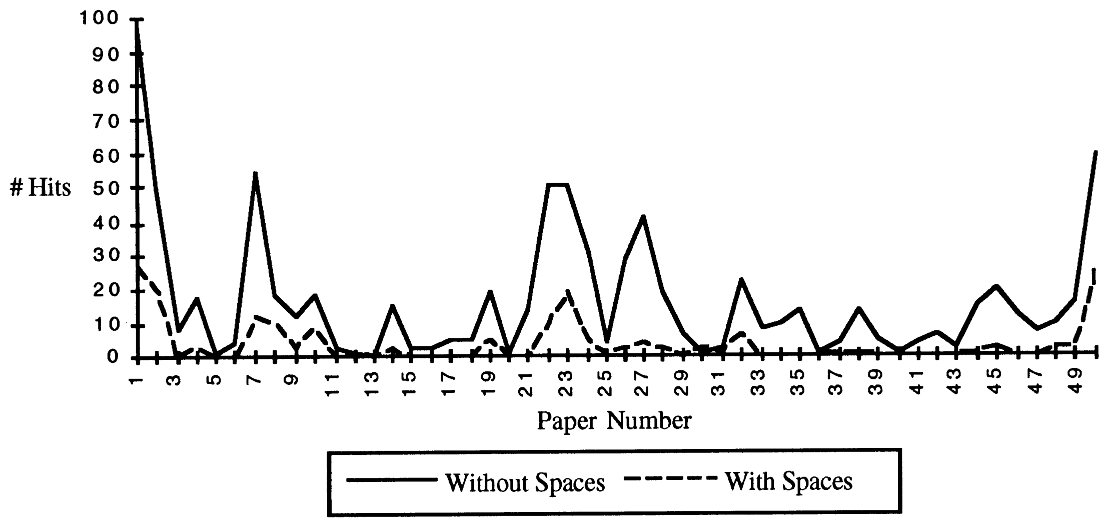
*Retrieval Results of the Neural Filter Based on NGrams with Spaces*

Similar simulations as above were done on the neural filter map trained with normal characters as well as spaces. By doing so, the map is also capable of expressing a correlation between word relations in the training and test text.

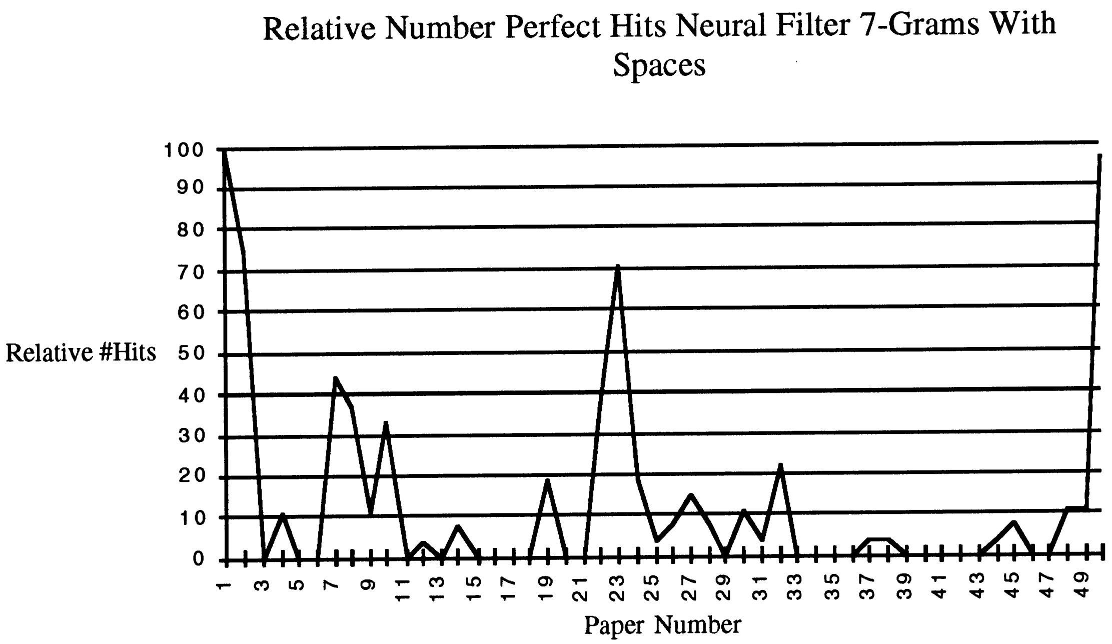
The number of possible n-grams held by the text increases dramatically as we incorporate spaces (without spaces a word of $m$ characters ($m>n$) holds $m-n$ different n-grams, with spaces a word of $m$ characters ($m>n$ or $m<=n$) holds $m+n-1$ different n-grams. Therefore, the size of the map must be larger compared to the case without spaces to remember the same amount of relevant n-grams. If the map is too small, too many n-grams will be bounced out.

The simulations were compared to the ones without spaces for the number of perfect hits (positive selection) as well as the error retrieval value (negative selection).

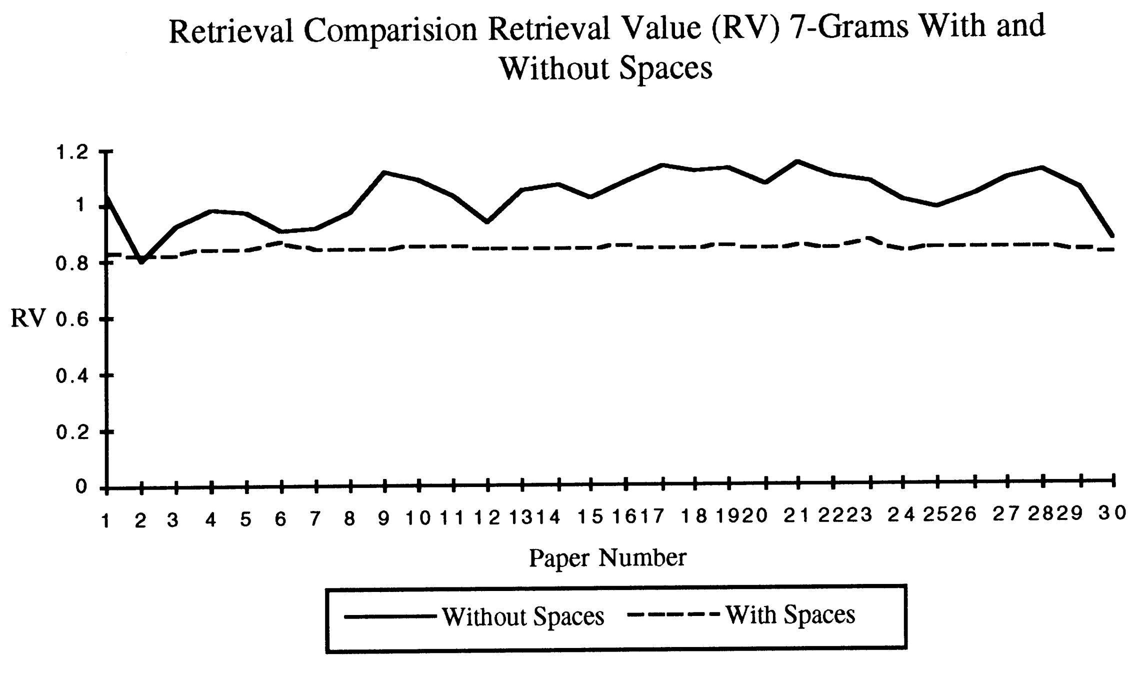
Retrieval Comparision Number Perfect Hits 7-Grams With and Without Spaces




In the case with spaces, the number of perfect hits is much smaller than in the case without spaces. At first sight this might look bad, but at second sight, the difference between the more and the less correlated papers is larger than before. Therefore, selection thresholds are easier to set. See the next graph for a plot of the number of hits with respect to the number of hits of the training text. The papers which are exactly on the same subject (and not just a little bit correlated) have a very high relative number of hits. Ones which are only a little related have a much lower retrieval value.




Below, the retrieval results of the negative selection functions of the algorithm with and the algorithm without spaces. Clearly, higher order n-grams only work in positive selection functions. Due to the high filtering effect (only very specific n-gram combinations are known and therefore recognized) and the normalization, almost all retrieval values are equal, even in the case of the 7-grams.



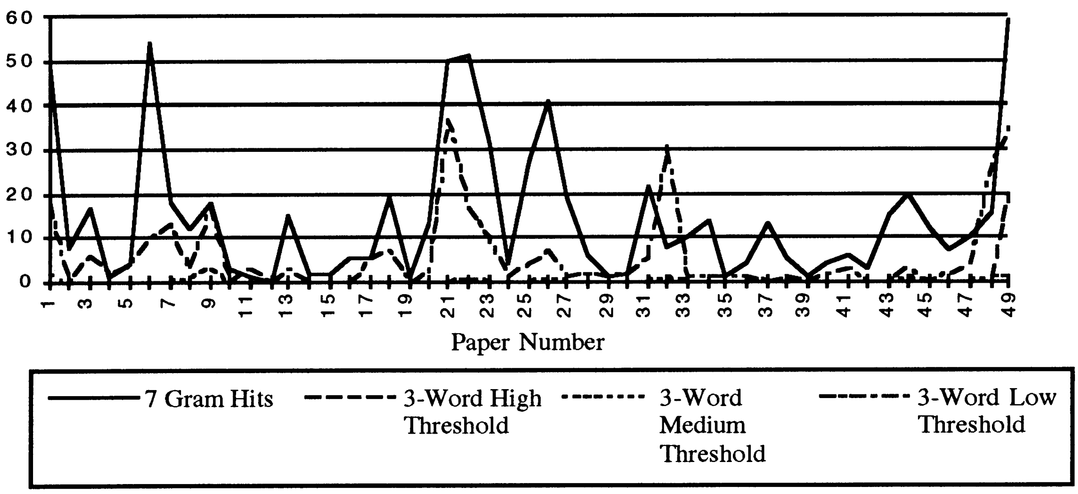
*Retrieval Results of the Neural Filter Based on Keywords or Large Preselected N-Grams*

The retrieval of n-grams on keywords has one tremendous advantage over the n-gram on characters: it is incredibly fast (40 MBytes Pravda in 5 hours on a PC). Because the system only knows 281 keywords (all the non-trivial words in the learning text), all other words don't have to be fed into the feature map, they are ignored. If on the other hand, proper word combinations are encountered, their retrieval value can be examined by feeding the word n-gram in the feature map. Due to the small amount of known keyword combinations, the threshold for a *perfect* match must not be too high.

In the following graph, the retrieval vales for the 7-gram characters and the 3-gram words are compared with each other. The 3-word retrieval is measured for three cases: a high, medium and low threshold. The high threshold resulted in one perfect match for the most correlated papers. All others equal zero. The medium threshold was a little better, but the low threshold worked best. One can see clearly that the 3-gram on words peaks less frequently than the 7-gram on characters, but if it peaks, it peaks high. This large difference between more and less related papers makes it easy to set a threshold.

This method filters sometimes too much, but if one really wants only the most correlated objects from a large amount of data, this method can do so in such a selection. Moreover, the same holds here as with the n-gram character filter with spaces: it should be used in combination with other, less strong filters to achieve a high quality filtering mechanism.

Retrieval Comparision Number Perfect Hits 3-Word Neural Filter



## Complexity Comparisons Neural Filter

The next graph summarizes the results in a quantative way. Beside the quality of the retrieval, the amount of computational power needed is interesting too. Especially since this research is in between statistical pattern recognition (fast) and symbolic linguistics (slow).

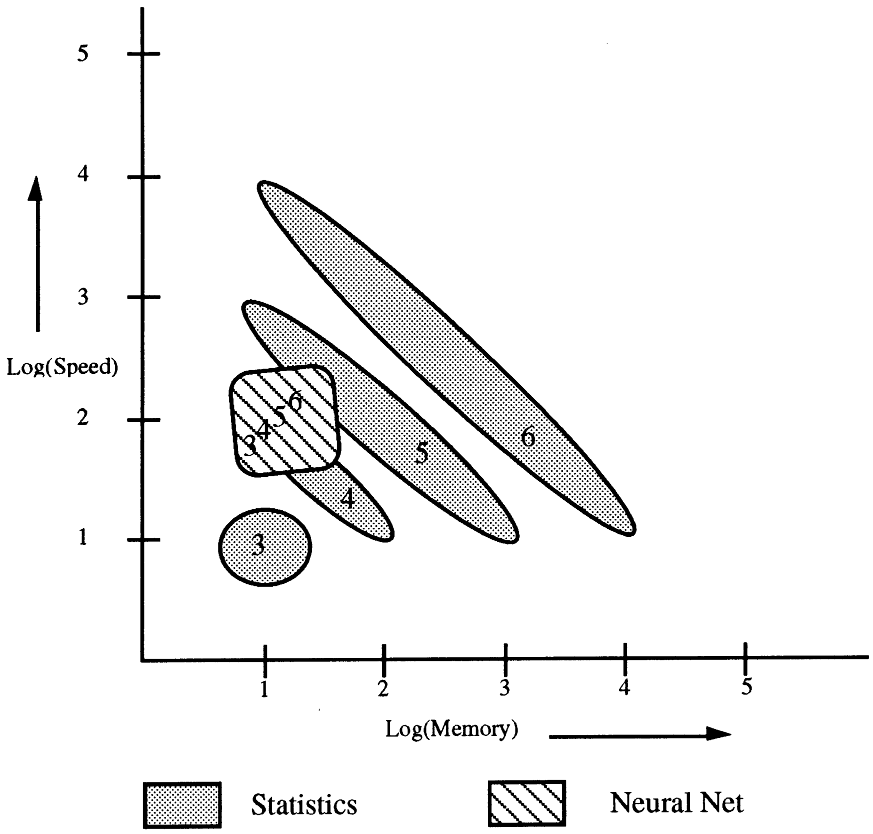
*Experimental Results: Derivation of Complexity*

Below, the graph holds an abstract representation of the experimentally derived complexity results (in time and space). The horizontal axis holds the logarithmic function of the amount of memory needed. The vertical one holds the logarithmic function of the required processor capacity. The numbers in the shadowed boxes represent the order of the Markov chains and the two colours represent the neural and statistical implementations. The long stretched circles from the statistical methods indicate an approximation of the implementation space for algorithms.

One can either a add large amount of memory or use direct addressing methods. Another method is to implement hashing techniques, ordering and generalization algorithms, etc., which take less memory, but probably more processor time (maybe not as drastically as indicated here, but there is definitely a trade-off which has comparable proportions as indicated in the graph). The form of the statistical areas is mainly due to the flexibility one has when implementing statistical algorithms.

The area of the neural implementation is much smaller, caused by the uniform data representation and the lack of procedural flexibility (i.e., making up a procedure to solve a local problem). However, the amount of memory and processor capacity needed increases considerably more slowly than with the statistics. For higher order problems the neural solutions are even faster, although all comparisons are based on simulations and *not* on parallel hardware (e.g., transputers or even *neural chips*).

Neural versus Statistical NGrams



*Theoretical Results: Mathematical Derivation of Complexity*

The graph presented in the previous section suggests an exponential complexity for the statistical n-gram algorithm and a linear complexity for the neural one. Here, the experimentally derived results will be proven correct by determining the complexity of both algorithms. In the statistical as well as the neuronal algorithm, the complexity is determined on basis of a serial implementation. The fact that the neuronal algorithm can be parallilized easily is ignored completely.

| | |
|---|---|
| N | Number elements (possible occurrences) |
| n | Order Markov chain (window size) |
| T | Number n-grams in training text |
| s | Number sensors per window |
| r | Number neurons in update region |
| L | Learning factor (the number of times the training set is passed) |
| p | Number of most frequent n-grams |
| $C_m$ | Memory complexity (space) |
| $C_c$ | Computational complexity (time) |

Used Symbols

Both algorithms filter the strings before they are processed. Because the statistical as well as the neural algorithm do this step, it is not interesting for a complexity comparison (except if it was the most expensive step in the calculation, which it isn't here). The following steps are important for the difference in complexity between these two types of algorithms. The statistical algorithm has to *count* the n-grams, *order* the table on frequency, and *normalize* the frequencies to probabilities before any query can be made. Thereafter, selections can be made by comparing a filter n-gram vector with the n-gram vectors of the free-text data base: the *retrieval* phase. The neural algorithm has two phases, a *learning* phase and a *retrieval* phase. By comparing the calculations steps and the memory needed, an insight in the quality of both algorithms can be obtained.

Two cases are being separated:

- All n-grams are being calculated, ordered and normalized. In the retrieval phase all n-grams are used.

- Only the *p* best n-grams are selected from the entire n-gram vector. These elements are being normalized and used in the retrieval phase.

The reason for this separation might be clear. In the first case, many elements equal zero. Therefore, a calculation based on the *p* best elements of the n-gram vector probably evolves to the same selection outcome. By using only the *p* best elements, the complexity can be reduced considerably. For both cases, the statistical as well as the neuronal complexity is calculated. First, we determine the complexity of the statistical algorithm in the case all n-grams are used in the calculation.

The complexity of the n-gram counting varies between two values. If one uses $N^n$ memory elements, every n-gram can be updated directly by addressing is as: $\Sigma(\text{char}_n)*26^n$. However, if *n* gets large, one definitely runs out of memory and a table holding the n-grams that occurred as well as their frequencies must be used. By ignoring the non-occurring n-grams, memory can be saved. The worst search algorithm in such a table is a binary one. Sophisticated hashing techniques probably result in a smaller value. Therefore the number of calculations needed in the counting phases shall be somewhere in between T and $T \cdot \ln(N^n)$. The Ordering phase is the most intensive one. If all n-grams are being ordered, the number of calculations needed in the Quick Sort algorithm is $N^n \cdot \ln(N^n)$. Finally the normalization phase uses $N^n$ steps. Every table element must be divided by the total number of counted elements. More advanced normalization methods may substitute the frequency of occurrence with a probability for each n-gram in a certain language, stipulating the non-frequent ones. The total number of calculations for the learning phase is the summation of the above mentioned values. In the box below, the result of this concatenation can be found.

The highest order term in this equation holds the complexity of the algorithm. All the other ones are negligtable in the long run. In other words, the ordering phase determines the complexity, the others are less important. Therefore, the complexity equals $O(x^n \cdot \ln(x^n))$. The retrieval phase has the same number of calculations increased with a term for the matching process. The complexity is of the same order: $O(x^n \cdot \ln(x^n))$. The amount of memory needed is somewhere between $q$ and $N^n$, where $q$ is the number of n-gram elements unequal zero. For large texts, $C_m$ probably reaches $N^n$.

| | |
|---|---|
| $C_{count}$ | $T < C_c < T \cdot \ln(N^n)$ |
| $C_{order}$ | $C_c = N^n \ln(N^n)$ |
| $C_{normalize}$ | $C_c = N^n$ |
| $C_{total\ learn}$ = | $C_{count} + C_{order} + C_{normalize}$ => |
| $C_{total\ learn}$ | $T + N^n \ln(N^n) + N^n < C_c < \ln(N^n) + N^n \ln(N^n) + N^n$ => |
| | $C_c = O(x^n \cdot \ln(x^n))$ |
| $C_{process\ text}$ | $T + N^n \ln(N^n) + N^n < C_c < \ln(N^n) + N^n \ln(N^n) + N^n$ |
| $C_{matching}$ | $C_c = N^n$ |
| $C_{total\ retrieve}$= | $C_{process\ text} + C_{matching}$ => |
| $C_{total\ retrieve}$ | $T + N^n \ln(N^n) + N^n + N^n < C_c < \ln(N^n) + N^n \ln(N^n) + N^n + N^n$ => |
| | $C_c = O(x^n \cdot \ln(x^n)) = O(x^n \cdot x^n)$ |
| $C_{memory}$ | $C_m = N^n$ => |
| | $C_c = O(x^n)$ |

Complexity for Statistical Algorithm: All N-Grams Used

If all n-grams must be ordered, a number of neurons between $q$ and $N^n$ must be used, where $q$ is the number of n-gram elements unequal zero. For large texts, this value reaches $N^n$, therefore, $N^n$ neurons are used in the determination of the complexity. In the learning phase, the number of calculations equals the times the learning set is passed, multiplied by the number of trigrams, multiplied by the number of calculations needed to learn one n-gram. This last term is determined by adding the calculations for the determination of the best element on the map and the update of the weights within on region. To compute the best neuron, $s \cdot n$ sensors of $N^n$

neurons must be evaluated. To update the weights, $s \cdot n$ sensors of $r$ neurons must be updated. Following from the equation in the box below, the complexity of the learn process is $O(x^n \cdot n)$. In the retrieval phase, the update term $r$ and the number of times the learning set passed the training procedure are eliminated, resulting in less calculations, but with the same complexity. The number of memory cells needed is larger than in the statistical algorithm. There ought to be a neuron for every possible n-gram. Every neuron has $n$ windows of $s$ sensors, resulting in a complexity of $O(x^n \cdot n)$.

| | | |
|---|---|---|
| # neurons = | $N^n$ | |
| $C_{learning}$ | $C_c = L \cdot T \cdot N^n \cdot s \cdot n + L \cdot T \cdot r \cdot s \cdot n$ | => |
| | $C_c = O(x^n \cdot n)$ | |
| $C_{total\ retrieve}$ | $C_c = T \cdot N^n \cdot s \cdot n$ | => |
| | $C_c = O(x^n \cdot n)$ | |
| $C_{memory}$ | $C_m = N^n \cdot s \cdot n$ | => |
| | $C_m = O(x^n \cdot n)$ | |

Complexity for Neural Algorithm: All N-Grams Used

Because the bigger part of the n-grams equals zero, it's quite silly to use all n-grams. The ordering phase is responsible for the largest part of the calculations. Therefore, this is the place to optimize our algorithms. By extracting the $p$ best n-grams, the number of calculations in the ordering phase is reduces to $N^n \cdot \ln(p)$; the number of elements times a binary search in an ordered best-n-gram table. The complexity value derived from the equation in the box then equals $O(x^n)$, a factor $n$ less. The same holds for the complexity of the retrieval phase. The amount of memory needed depends on the method used in the counting phase. This value varies between $p$ and $N^n$.

| | | |
|---|---|---|
| $C_{count}$ | | $T < C_c < \ln(N^n)$ |
| $C_{order}$ | | $C_c = N^n \ln(p)$ |
| $C_{normalize}$ | | $C_c = p$ |
| $C_{total\ learn}$ | = | $C_{count} + C_{order} + C_{normalize}$ => |
| $C_{total\ learn}$ | | $C_c = T + N^n \ln(p) + p < C_c < \ln(N^n) + N^n \ln(p) + p$ => |
| | | $C_c = O(x^n)$ |
| $C_{process\ text}$ | | $C_c = T + N^n \ln(p) + p < C_c < \ln(N^n) + N^n \ln(p) + p$ |
| $C_{matching}$ | | $C_c = p$ |
| $C_{total\ retrieve}$ | = | $C_{process\ text} + C_{matching}$ => |
| $C_{total\ retrieve}$ | | $C_c = T + N^n \ln(p) + p + p < C_c < \ln(N^n) + N^n \ln(p) + p + p$ => |
| | | $C_c = O(x^n)$ |
| $C_{memory}$ | | $p < C_m < N^n$ => |
| | | $1 < C_m < O(x^n)$ |

Complexity for Statistical Algorithm: *p* Best N-Grams Used

However, in the neural algorithm, the number of used n-grams is determined by the number of neurons in the map (every neuron can hold up to one n-gram). All less frequent n-grams are absorbed by the more frequent ones. If *p* neurons are used, the number of calculations needed in the learning phase reduces dramatically to $O(n)$. The same holds for the retrieval phase. The amount of neurons needed is of the order *n*.

| | | | |
|---|---|---|---|
| # neurons | = | p | |
| $C_{learning}$ | | $C_c = L \cdot T \cdot p \cdot s \cdot n + r \cdot s \cdot n$ | => |
| | | $C_c = O(n)$ | |
| $C_{total\ retrieve}$ | | $C_c = T \cdot p \cdot s \cdot n$ | => |
| | | $C_c = O(n)$ | |
| $C_{memory}$ | | $C_c = p \cdot s \cdot n$ | => |
| | | $C_m = O(n)$ | |

Complexity for Statistical Algorithm: *p* Best N-Grams Used

Below, the various complexity values are combined in a table. In the brute force comparison algorithms (the statistical as well as the neuronal), the order of calculations is exponential with respect to the window size. The neural algorithm is a bit better, but uses a little more memory than the statistical one. Therefore, it will be hard to increase the context sensitivity of such systems. On the other hand, if only the '*p* best n-grams' algorithms are used, the statistical algorithm has an $O(x^n)$ complexity, where the neural one has an $O(n)$. This is an exponential versus a linear complexity, both with an up to exponential and a linear memory usage. Here, the neural algorithm definitely outperforms the statistical one for large *n*'s ($\geq 4$). Please be aware that the neural complexity calculations are based on serial simulations and *not* on parallel ones. If the Kohonen feature maps would be implemented in large neural chips, the results would be even better [12].

| | Complete Sort | | | | Select *p* best n-grams | | | |
|---|---|---|---|---|---|---|---|---|
| | Learning | | Retrieval | | Learning | | Retrieval | |
| | Speed | Mem | Speed | Mem | Speed | Mem | Speed | Mem |
| Statistics | $O(x^n \cdot n)$ | $O(x^n)$ | $O(x^n \cdot n)$ | $O(x^n)$ | $O(x^n)$ | $< O(x^n)$ | $O(x^n)$ | $< O(x^n)$ |
| Kohonen Net | $O(x^n \cdot n)$ | $O(x^n \cdot n)$ | $O(x^n \cdot n)$ | $O(x^n \cdot n)$ | $O(n)$ | $O(n)$ | $O(n)$ | $O(n)$ |

Moreover, one has to realize that these results are strictly theoretical. They only hold in the long run. In the beginning, the statistics outperform the neural algorithm, but as *n* grows, the neural algorithm surpasses the statistical one. This is mainly due to the sorting phase in the statistical algorithm. One can optimize this, but the complexity remains exponential, were the neural algorithm remains linear.

[12] Moreover, by using methods as proposed by [Kelly, 1991] and [Koikkalainen et al., 1989] the search for the Best Matching Unit (BMU) can be done with the aid of a binary tree, storing the weight vectors in an ordered fashion. Then, the complexity in speed decreases from $O(n)$ to $O(\ln n)$. The complexity in space increases from $O(n)$ to $O(n \cdot \ln n)$.

## Simulations and Results Neural Interest Map

A vector representing text distribution features can be derived for every paper in the data base. Such a vector then represents a fingerprint of a data base object. Fifty papers were scanned en their corresponding vectors were taughed to the neural in the following simulations. The Kohonen learning mechanism is just standard. No special features were used. Two aspects are pointed out before the simulations are discussed. First, these simulations differ from the ones as proposed by [Gersho, 1990a-b], [Wermter, 1991] and [Lin et al., 1991] in that the former use very restricted text parts for the derivation of the feature vectors (mostly titles) and that the methods are based on a custom well optimized hand-made keyword selection. Here, the keywords are derived from the text in the objects automatically and the feature vectors are based on keyword distributions in the learning text. Therefore, the vectors have very high dimensions and must be trained for long times. The result is a fully automatic (neural) clustering mechanism.

The Neural Filter simulations could still be implemented on a high end PC. These Neural Interest Map required a more powerful computational basis. This was due to several reasons. First the simulations needed many more training cycles. These learning cycles on their own took longer because they were based on larger vectors (500 to 2500 dimensional). However, the main reason why the PC was no longer suited for the simulations was that the PC could not calculate with large enough precision to garantuee convergion. One definitely needed the extended precision calculations of the Sun IPC to organize the elements in the map based on very small differences in vector dimensions. Sometimes, even the Sun floating point were not good enough, and the map could not converge. This was especially the case with the n-gram based simulations.

*Preprocessing Keywords and N-Grams*

Before the vectors can be taught to the neural map, they have to be derived from the free-text data base in the first place. This can be done by some preprocessing programs. As an initial step we derive the *m* most frequent words used in *all* the text parts. Next, the word distribution of these *m* words in the *n* text parts must be calculated. The distribution can be expressed in various forms:

- The occurrence can be measured (0 equals no occurrence, 1 equals the keyword occurs, the number of times it occurs is ignored).
- The word frequency can be normalized with respect to the total number of keywords in all text parts.
- The word frequency can be normalized with respect to the maximum occurrence of this specific keyword in all the text parts.
- The word frequency can be normalized with respect to the total occurrence of this specific keyword in all the text parts.

Once the *n* vectors of *m* dimensions have been derived, they are taught to the neural net in a random way. After a certain training time, the neural net holds a representation of the relations between the papers in the data base. Related papers shall be stored in neighbouring neurons.

When we substitute the keywords by n-grams (with spaces), a language independent clustering method results. First, the *n* most frequent n-grams are determined. Next, the distribution of these *n* trigrams in *m* text parts is examined. The obtained vectors can be normilized in the same different ways as the keywords are (see above). The n-gram vectors are much larger than the keyword vectors. In general, we taught only those n-gram dimensions that were unequal zero. To increase performence, the number of n-grams can be reduces even more. However, this can limit the cluster information. The normalized n-gram distribution vectors (one per text part) are taught to the Kohonen feature map in random sequence. After learning,the map holds related papers in neighbouring areas.

As with many computational linguistic problems, the solution of the clustering problem lies in the proper choices of the data representation.

*Results Interest Map Based on Keywords and Large Preselected NGrams*

In the simulations of the Neural Interest map based on keyword frequencies, a 10 by 10 map with 500 input sensors per neuron was used. The maps were trained somewhere between 5.000 up to 15.000 train cycles.The following maps were found:

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 0 | 2 | xxxxxxx | 23 | xxxxxxx | 7 | 9 | 11 |
| 0 | 0 | 4 | 3 | xxxxxxx | 1 | 35 | 12 | xxxxxx | xxxxxxx |
| xxxxxxx | xxxxxxx | xxxxxxx | 20 | 20 | 47 | 47 | 35 | 36 | 32 |
| 21 | 21 | 22 | 20 | xxxxxxx | 34 | xxxxxxx | 45 | 45 | 28 |
| 21 | 26 | xxxxxxx | xxxxxxx | 44 | 39 | 39 | 45 | 31 | 32 |
| 27 | 25 | xxxxxxx | 24 | xxxxxxx | 39 | 39 | 31 | 31 | 31 |
| 29 | xxxxxxx | 19 | xxxxxxx | 30 | xxxxxxx | 49 | 41 | 38 | 48 |
| 29 | 15 | xxxxxxx | xxxxxxx | xxxxxxx | 42 | 40 | xxxxxxx | 46 | 38 |

10 by 8 Interest Map of 500 Keywords: Normalized With Respect to Total Number of Words

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 4 | 4 | 46 | 20 | 20 | xxxxxxx | 13 | xxxxxxx | 6 | xxxxxxx |
| xxxxxxx | 46 | 46 | 46 | 20 | 5 | 13 | xxxxxxx | xxxxxxx | 2 |
| 41 | 41 | xxxxxxx | xxxxxxx | xxxxxxx | 5 | xxxxxxx | 45 | 45 | xxxxxxx |
| xxxxxxx | 8 | 8 | xxxxxxx | 31 | xxxxxxx | 32 | xxxxxxx | xxxxxxx | 1 |
| 29 | 29 | xxxxxxx | 17 | xxxxxxx | 49 | xxxxxxx | 28 | xxxxxxx | xxxxxxx |
| xxxxxxx | 0 | 0 | xxxxxxx | 48 | xxxxxxx | 37 | xxxxxxx | 18 | xxxxxxx |
| 23 | 23 | xxxxxxx | 12 | xxxxxxx | 35 | xxxxxxx | 42 | xxxxxxx | 27 |
| 23 | 34 | 34 | xxxxxxx | xxxxxxx | 40 | 38 | xxxxxxx | 22 | 7 |
| xxxxxxx | 26 | 34 | xxxxxxx | 47 | xxxxxxx | 44 | 25 | 19 | 10 |
| 26 | 26 | xxxxxxx | 43 | xxxxxxx | 33 | xxxxxxx | 30 | 21 | 16 |

10 by 10 Interest Map of 500 Keywords: Normalized With Respect to Total Occurrence of Word in Specific Dimension

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 31 | xxxxxxx | xxxxxxx | xxxxxxx | 18 | xxxxxxx | xxxxxxx | 8 | xxxxxxx | 9 |
| 32 | 30 | xxxxxxx | 20 | xxxxxxx | xxxxxxx | 19 | xxxxxxx | 7 | xxxxxxx |
| 34 | xxxxxxx | 23 | xxxxxxx | 22 | 21 | xxxxxxx | 6 | xxxxxxx | 3 |
| xxxxxxx | 25 | xxxxxxx | 24 | xxxxxxx | xxxxxxx | 43 | xxxxxxx | 5 | 4 |
| 33 | xxxxxxx | 26 | xxxxxxx | 44 | 42 | xxxxxxx | 45 | xxxxxxx | 1 |
| xxxxxxx | 49 | xxxxxxx | 46 | xxxxxxx | xxxxxxx | 41 | xxxxxxx | 2 | xxxxxxx |
| 47 | xxxxxxx | 48 | xxxxxxx | 37 | xxxxxxx | xxxxxxx | 40 | xxxxxxx | 0 |
| 47 | 47 | xxxxxxx | 38 | xxxxxxx | 35 | 36 | xxxxxxx | 39 | xxxxxxx |

10 by 8 Interest Map of 500 Keywords: Normalized With Respect to Maximum Occurrence of Word in Specific Dimension

Although the above three maps hold some interesting relations, the overall conclusion is that they are quite wrong. The papers zero, one and fourty nine are all on *nuclear weapons* *restriction talks*. Zero and 1 are in neighbouring regions, but 49 never is. The papers 25, 32, 34 are all on economical issues, etc. The vectors used were too much related due to the small values caused by the normalizations used. The coding that did work well was the one where each dimension represented one word. If a word occurred in the text part, the dimension equals one, otherwise it equals zero. By eliminating the frequency of occurrence, the vectors became more distinguishable and therefore better learnable. The picture on the next page shows the map obtained after 15.000 training cycles.

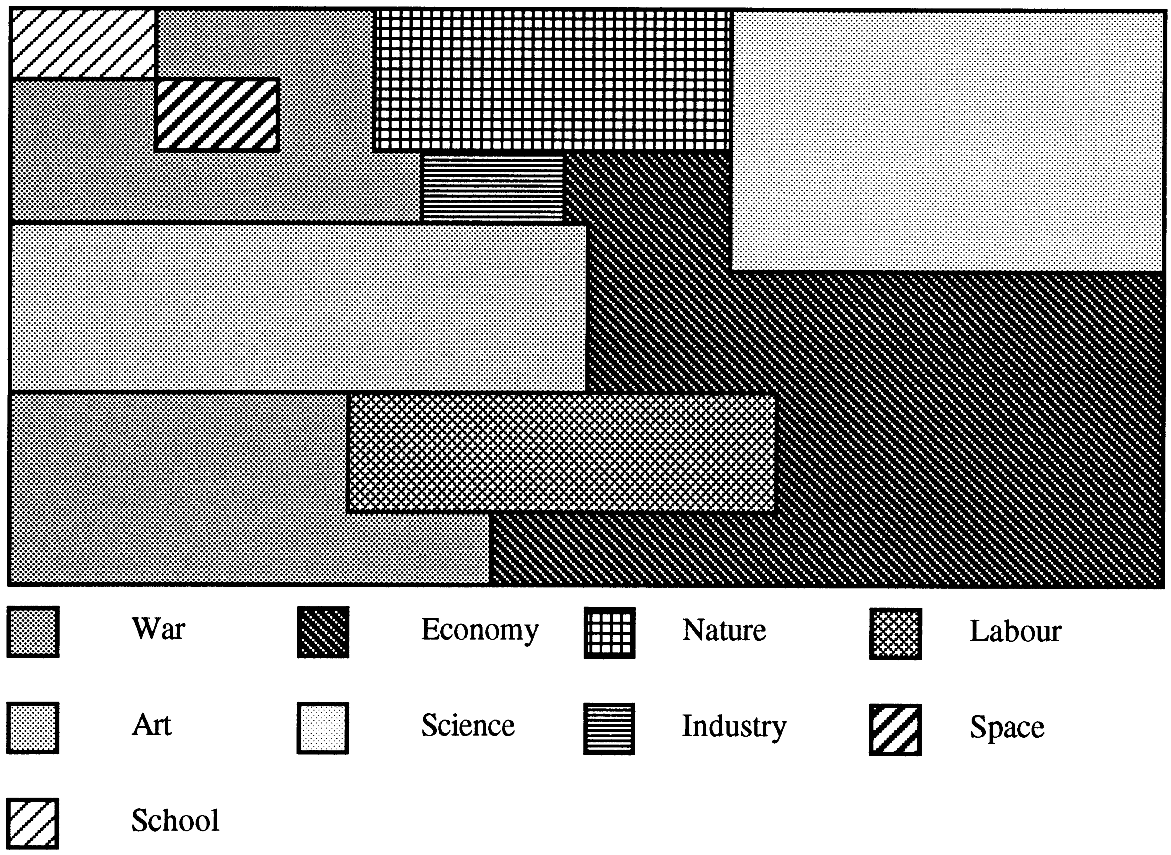
| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 20 | xxxxxxx | 22 | 21 | 24 | xxxxxxx | 48 | 48 | xxxxxxx | 47 |
| xxxxxxx | 18 | xxxxxxx | 19 | xxxxxxx | 46 | xxxxxxx | xxxxxxx | 49 | xxxxxxx |
| 23 | xxxxxxx | 35 | xxxxxxx | 38 | 37 | xxxxxxx | 40 | xxxxxxx | 0 |
| 23 | xxxxxxx | xxxxxxx | 33 | xxxxxxx | xxxxxxx | 39 | xxxxxxx | 41 | xxxxxxx |
| xxxxxxx | 36 | xxxxxxx | xxxxxxx | xxxxxxx | 34 | xxxxxxx | 42 | xxxxxxx | 1 |
| 2 | xxxxxxx | 12 | 11 | 10 | xxxxxxx | 44 | xxxxxxx | 43 | xxxxxxx |
| 3 | xxxxxxx | 13 | 14 | xxxxxxx | 17 | xxxxxxx | 45 | xxxxxxx | 32 |
| xxxxxxx | 4 | xxxxxxx | 15 | xxxxxxx | xxxxxxx | 16 | xxxxxxx | 31 | xxxxxxx |
| 5 | xxxxxxx | 7 | xxxxxxx | xxxxxxx | 26 | xxxxxxx | 29 | xxxxxxx | 30 |
| xxxxxxx | 6 | 9 | 8 | xxxxxxx | xxxxxxx | 25 | 27 | 28 | xxxxxxx |

10 by 10 Interest Map of 500 Keywords: 01 Vectors: 0: Word does not occur, 1 Word Occurs

Translated to what the areas mean (by using the table with short document descriptions), the neurons are related to the following objects:

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Smoking | | N Weap | N Weap | Africa | | Genetic | Genetic | | Justice |
| | Space | | Nature | | Quality | | | SDI | |
| Afganist | | Baking | | Theatre | Old Past | | France | | SDI |
| Afganist | | | World | | | Cambodia | | Santa | |
| | Nature | | | | Working | | Sports | | New Year |
| Museum | | Poem | Rabbits | Story | | N Weap | | Indust | |
| Literat. | | New Year | Labour | | Com. Pty | | Gasoline | | Economy |
| | Art | | Constr. | | | Transp. | | Soc. | |
| War | | Peace | | | Dreams | | Carnival | | Cars |
| | N. Weap | N. Weap | New Year | | | Economy | Ec.Mil. | Ec.Italy | |

There are two clusters related to war. One on the bottom left which holds documents on conventional warfare. The upper left cluster is based on the more scientific SDI warfare. Why these two clusters are so far separared is not clear. On the right side, a large cluster with economical documents can be found, Within this group, smaller neigbourhoods hold transportation cluster, socialism clusters and quality (a hot item in the USSR) clusters. The more manual labor based economical documents are on the left side. Surrounded by a cluster for Art and one for Nature (e.g., Africa). This can be translated to the following global area topics:




Although some groups are still separated for reasons which are not clear yet, the overall impression is that this map holds the semantical relations between the documents.

*Results Interest Map Based on Trigrams*

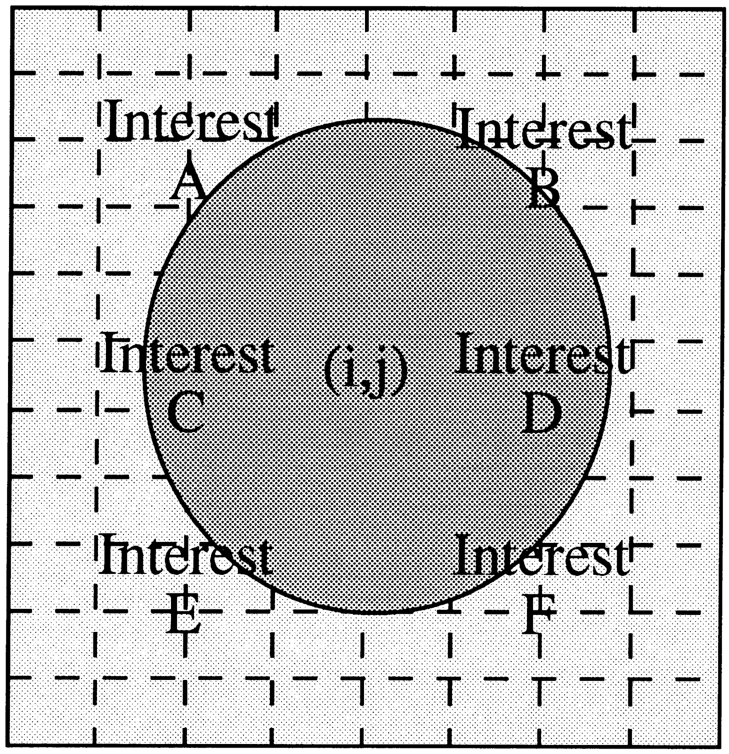
The words in the simulations just mentioned were derived automatically, so the step to n-grams isn't that large. By incorporating spaces, relations between words can be characterized by n-grams. Because the system cannot learn to much n-grams, the *m* most frequent n-grams can be taught only. This set of n-gram vectors must be derived first, resulting in an exponentially complex problem with respect to the size of *n*. Therefore, only simulations for *n* is 3 are carried out. Because the differences between the vectors were very small and the vectors very large, one cannot use the real frequencies of trigram occurrences in the training set. Therefore, the real frequencies were substituted by 0 and 1's, as in the simulations above. The map formed organized itself after long training times. The results are comparible to the keyword based organizations, although it takes much longer to derive them.

*Retrieval Neural Interest Map*

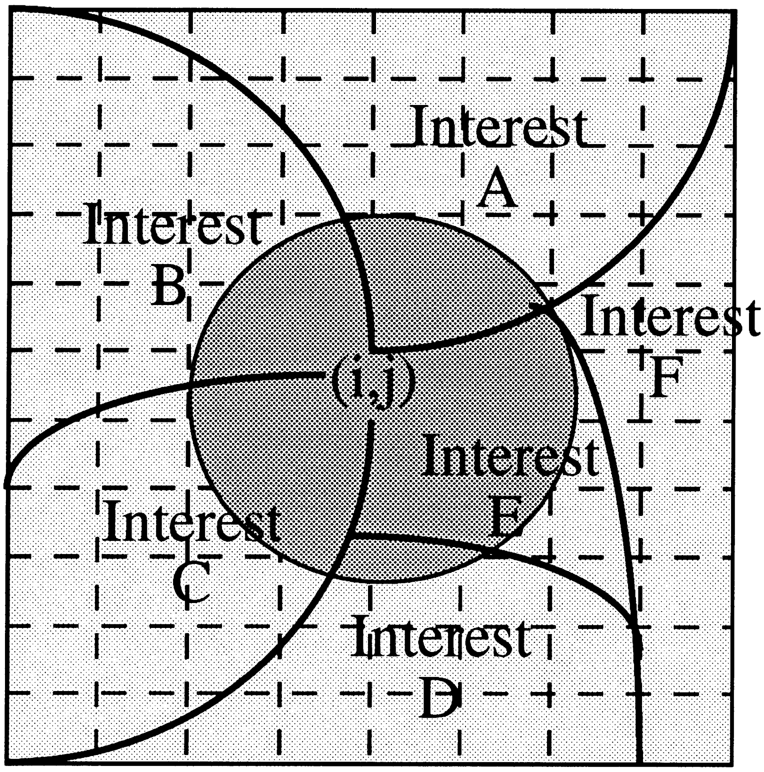
The retrieval is always based on the calculation of a keyword or n-gram vector with classical methods. The keyword mathc is quite fast, because one only needs to calculated the frequency of known words. Normally, the distionary is of restricted size. By incorporating advanced hashing techniques, the elements can be updates quickly.

More difficult is the derivation of a n-gram vector per document. Although one also has a restricted feature space, it is much larger than the keyword space. Moreover, search times are longer for n-grams than for keywords (see Neural Filter results for a motivation).

Once the vector has been determined, it can be fed forward in the feature map to derive the Best matching Unit (BMU). This BMU represents the document most correlated with the vector. The neurons within a certain euclidean distance hold related documents. However, one needs the BMU as well as the cluster boundaries to make a responsible desicion on the measure of correlation between documents. An example is given in the picture below.



If the vector **x** correlates best to the BMU with weight **w** at neuron **(i,j)**, then all neurons in positions ( **i±δ, j±δ**) are supposed to be related (dark circle). If we have a map as indicted above, all documents within a certain euclidean distance are related. The neuron **(i,j)** is then related to all interests in the same way. However, even if we take δ small, this is not always thruth. What if the neuron at the feature map looks like the one derived in the Interest Map. If the BMU seems to be the neuron at position **(i,j)** (see picture below).




The BMU is positioned exactly at the border of multiple interests. The reason why these interests are neighbouring is not because they are related, but because they are forced to interconnect due to the dimension reduction properties of the feature maps. If we use the euclidean distance as a selection criterion, the documents selected are not the proper ones. Some interests which are neighbouring have nothing to do with each others.

So, the only way to make a reasonable decision is by incorperating the BMU as well as the cluster boundaries. This is a big disadvantage, because then one has to determine the cluster boundaries manually. These desicions take much work, are very personal and therefore subjective and sensitive to errors [13].

More on this problem can be found in the next sections.

---

[13] Currently, the possible automatic determiniation of cluster boundaries is under investigation. By measuring the distance between the weights of two neighboring neurons (the Euclidean as well as the Cosine), a feature map landscape can be calculated which holds high values for non-related neighboring neurons, also called fractions in the homogenuity of the distribution. These hills might then be incorporated as clustrer boundaries in the selection process to avoid selection of non-related objects.

## Discussion

This section discusses several items in the field of neural information retrieval. To get a better overview, the subjects are organized under different headings.

*Neural Nets for Information Retrieval & Information Retrieval for Neural Nets*

Information retrieval, being a clear pattern recognition problem, has mainly been benefiting from statistical pattern recognition technologies. The enormous amounts of data to be processed actually didn't allow any other methods within practical limitations. As time passed by, many researchers tried to increase the level of analysis without blowing up the computational needs. Due to this constraint, the information retrieval tool box could never use any linguistic theory. Therefore, the used algorithms are restricted to local surface analyses.

Recent research in connectionist natural language processing showed interesting self-organizing models that can learn finite state grammars and simple semantical relations from unformatted data. Moreover, these neural devices showed remarkable competetance in clustering and classification tasks of incomplete data sets. All these properties are well known functional demands for information retrieval systems. This combined with the implicit parallelism makes one wonder why the research toward neural nets in information retrieval still is restricted to such a small school, although the number of papers appearing in literature is increasing [Belew, 1986], [Personnaz et al, 1986], [Doszkocs et al., 1990], [Gersho, 1990a-b], [Kwok, 1990], [Kwok, 1991], [Rose, 1990], [Allen, 1991], [Scholtes, 1991d-g & 1992a-c], [Wermter, 1991], [Lelu, 1989, 1991], [Hingston et al., 1990], [Wettler, 1989,1990], [Mozer, 1991], [Rapp, 1991], [Bochereau Laurent et al., 1991], [Brachman et al., 1988], [Eichman et al., 1991, 1992], [Jung et al., 1991], [Van Opdorp et al., 1991].

Beside the positive contribution from neural nets to information retrieval, there is also a return. Information retrieval has a long and well known history in statistical pattern recognition. Many problems have indeed been solved by using such methods. Comparisons of such results with new results in neural information retrieval give us an entry to gain a much better insight in the exact relation between neural nets and other classical pattern recognition solutions. Because, if neural nets are such good pattern classifiers, where does one position it with respect to the known pattern recognition theories.

Even more interesting is the contribution of information retrieval to NLP. Because information retrieval problems are often much simpler, they clarify neural bottle-necks much easier than NLP problems, thus contributing to the development of better neural models for NLP.

*Information Retrieval With N-Grams*

A major drawbacks of a keyword-matching system is the need for a dictionary, and the lack of context. Markov chains over words solve the latter, but need enormous amount of memory (at least in statistics). N-grams seems to provide an acceptable solution for a reasonable price. They are not sensitive to noise (see examples), they can be used without predefined linguistic knowledge, and they can be derived automatically.

By varying *n* between 3 and 7 a proper contextual analysis is derived without the usage of dictionaries. If one incorporates spaces in such a higher order model, n-grams with starting and trailing spaces hold information on words transitions, thus implementing syntactical and (low level) semantical relations [Brown et al., 1990], [Jelinek, 1989 & 1991a-c].

Here, the n-grams are used in the filtering as well as the clustering algorithm. Especially in the filtering solution do they do very well. The ability to represent 7-grams without any computational pain definitely improves recognition rates. The n-gram over n-gram simulations showed even better results. On the other hand, the application of n-grams in the clustering algorithm wasn't that easy. It needed more computational power than expected. However, by using various optimization techniques, the results were easier derived than if the simulations had to be hand-coded.

*Statistical versus Neural Information Retrieval*

Normally, statistics is faster than neural nets. Here, some opposing results are found in the case of the neural filter. Where the statistics are either exponential in time or space, the neural algorithm scales linear. If one wants to use higher order n-gram analyses to avoid dictionaries, suffix preprocessing, and other language dependencies, linear scalability is an essential property. The most obvious statistical algorithm does not fulfil this demand.

Recent claims that all neural net solutions can be implemented with regular statistics at higher speed and with more flexibility, seem true at first sight. But, there are cases in which the statistical solution loses the competition is lacking, as shown by the complexity comparisons in the neural filter experiments.
The flexibility one has in statistical models (local optimization) lacks in neuronal modelling. However, once an efficient neural algorithm has been developed, it can be scaled smoothly to much larger dimensions of the problem. Moreover, neuronal models have many implicit properties, such as inherently built-in generalization and association, which one must define explicitly (the draw back of flexibility) in statistical ones. So even if one constructs an algorithm that is as efficient as the neural one, then one still has to hand code many functions that neural nets already provide.

Please note again that the complexity calculations are based on serial simulations only. However, the fact that the neural algorithm can be parallelized efficiently, should not be forgotten. All Neural Filter simulations are implemented on a 386 PC with 600 Kbytes (available) RAM.

*The Filter and Interest Map as Hashing Functions and Semantic-Cognitive Maps*

In this paper, algorithms from two different perspectives have been presented. The neural filter algorithms learn a (static) query and match it against a dynamic text. This algorithm outperforms the statistical one in speed and quality.

Which selection rule suites best in a certain application depends much on the person using it. The n-gram on character filters is capable of filtering large amount of data, but cannot specifically indicate which ones are the best. The n-grams on words and the models that incorporate spaces can discover high correlations within times equal (or even faster) to the low level n-gram filters, but eliminate all non so closely related papers. Most of the time, combinations of such selection functions work best: negative to filter the noise and positive to indicate possible candidates.

If one studies the behaviour of the Neural Filter, the question arises what the exact relation with another well-known addressing technology is: *hashing*. On the one hand, the relation is very clear: neural nets are large (calculating) associative memories, able to store elements efficiently. On the other hand, the reason why certain elements are stored and others are eliminated is not clear yet [Boyer et al., 1977], [Bozinovic et al., 1982], [Harrison, 1971], [Knuth et al., 1977], [Larson, 1988], [McIllroy, 1982].

The Interest Map stores various n-gram and vectors from more static text and matches this neural map against (dynamic) queries. Here, the use of neural nets does not have the same amount of success as the neural filter, although the latter is a very well generalizer. Future research should determine whether statistics can be outperformed here as well .

The relation between the Neural Interest map and well known semantical and cognitive maps obvious in a certain way. However, the Kohonen feature map is just a small step in the direction of the functional requirements as ment in literature. One should not overestimate the power of feature maps. Especially the problems related to the cluster boundaries (due to the dimension reduction) are quite hard to solve.

*Higher Order Linguistics & Knowledge Representation*

A main problem of information retrieval has been the dilemma that higher level analyses were not available at reasonable prices. Recent research in connectionist neural nets showed how to learn complicated finite state grammars with recurrent neural nets. By learning word sequences, neural nets were able to learn regular grammars. Although these grammars are the simplest possible, they can definitely increase recognition performence. However, the infinite Markov models learn very slow and are quite unstable. But, it seems that most information retrieval systems are aided sufficiently with a restricted Markov model (such as trigrams over words). The fact that neural nets can learn these relations in linear time, opens a scala of new possibilities for neural nets in information retrieval.

So much for the treatment of structural analysis in IR. Another important problem is to incorporate meaning in IR. Yet, there is no real meaning involved in the algorithms proposed. There are the contextual relations incorporated in the n-gram representation. By generalizing over these contextual structures, simple semantic relations can be derived. However, real meaning and the interpretation of conceptual structures is something more complicated, and seems not to be taken lightly or solved solely by means of n-grams.
Synonyms can be added to a neural filtering model by comparing the input text with some synonym networks, resulting in a synonym group. Then, this group can be compared to the query. The other way around is also possible for the Interest Map. The derivation of the synonym groups can be done completely unsupervised from flat strings with e.g., the algorithm proposed in [Scholtes, 1991a-c], where substitutional (semantical) identical words are clustered in neighbouring regions of a recurrent Kohonen map. Of course, there are also various other mechanisms available to solve this problem. This also indicates an answer to the problem of meaning. What kind of meaning do we want: complex logic, or simple generalizations? The addition of microfeatures can probably solve the largest part of the problem. But they introduce many new problems. Therefore, these low level semantics seem to be a better alternative for the time being.

Other research focuses on knowledge representation structures for information retrieval. The early connectionist models were mainly used for such applications. Only recently have neural nets been used for clustering tasks. A possible use of such clustering neural nets for knowledge representation is in the use of hierarchical feature maps, where relations between objects and classes of objects is catched in the hierarchical structures. Another solution can be found in [Allen, 1991], who uses a simple recurrent network (SRN) to teach semantical issues to a back-propagating net.

*Problems of Feature Maps*

There are some serious problems with the feature maps as used in these simulations. First there is the neural filter. We do use this property of the feature map during the learning phase, but it results in strange and sometimes unwanted effects. Frequent n-grams disappear and non-existing ones appear. However, if we eliminate the neighbourhood effects and thus implement a form of Principle Component Analysis (PCA) [Oja et al., 1988, 1991], the non-frequent n-gram are no longer thrown out of the feature map [14]. If we reduce the neighbourhood effects, the model converges much slower and ends up representing the n-gram distribution less well[15]. So even if we don't explicitly use them (or don't exactly understand them), the neighbourhood effects seem to have a significant role.

Next, how do we interprete a topological map of n-grams or keywords. We do not use this property of Kohonen feature maps in the retrieval phase because we do not know how! This problem is closely related to some assumptions we made about the underlying probability distribution. The Kohonen Feature Map requires a predefined network structure (e.g., fixed dimension, fixed rectangular or hexagonal connection structure and fixed square, triangular, circular but continuous homogeneous topology). If we consider the map used in these simulations we do in fact presume that language is two-dimensional, rectangular and homogeneously distributed, which is of course not true.

This problem is even more clear in the case of the interest map. After the training phase, related objects must be in related neighbourhoods. However, what if a paper is on the border of multiple clusters. If this neuron is selected as the Best Matching Unit (BMU) on the Kohonen feature map, then the Euclidean distance does not represent a proper measure of correlation. One has to incorporate the cluster boundary knowledge in the classification decision. Such cluster boundaries must me derived by the model itself and not by an external supervisor.

Future research concentrates on the development of learning algorithms which automatically construct feature maps that do fit the underlying probability distribution. These algorithms should develop a feature map in an n-dimensional space with various interconnecting schemes and form different maps representing clusters of interest [Fritzke, 1991a-b] [Martinetz et al., 1991]. These maps can then be combined in a hierarchical network which models the interest of an individual automatically from free text.

*Adaptation*

In this paper, the Kohonen map used is forced to converge by a decreasing region size and is therefore not adaptive to a slowly changing environment. By changing the learning rule, better learning functions can be developed, resulting in an adaptive neural model of someone's interest that changes slowly in time.

---

[14] According to Professor E. Oja, the underlying distribution function of the n-grams is much too clustered to use PCA's or comparable methods. Such mechanisms only work properly for very homogeneous data sets of noisy natural data.

[15] According to Professor T. Kohonen, it is really difficult to understand what is happening on the feature map. However, if the neighbourhood function has been eliminated in other applications, learning slowed down and the cluster boundaries on the feature maps were much more discontinuous. So, even if you don't understand the topological map, you can still use the neighbourhood effects to to end up with a smooth representation of the probability function of the learning set.

*Kohonen Feature Maps, Back Propagation and Other Neural Paradigms*

Is the Kohonen feature map the best neural model for the simulations carried out? There are many other neural models. The early neural information retrieval used localist knowledge representations (one neuron for one concept). Recent efforts showed the application of feed-forward and recurrent back-propagating nets. Kohonen feature maps are just recently invoked in IR applications. Hopfield nets and other associative memories have also been used, but only rarely.

In general, clustering and generalization problems are best solved with self-organizing nets, such as the Kohonen feature map, the Simple Recurrent Network (SRN), and ART. Mapping problems or function approximations can best be done by a feed-forward back-propagating neural net. Temporal processing can best be done by an SRN or any other recurrent model. Associative memory problems might be solved be either neural net: BP, Kohonen, ART, etc. Of course, these applications can also be solved with other net types, but the nets mentioned are the most natural choices.

Information Retrieval is a clustering problem. Based on a selection of specific features (e.g., n-grams or keywords), a representation of an object is derived by feature extraction. The objects are categorized in clusters by the retrieval function. Main issue is the determination of such features, so the difference between clusters is as big as possible (or, as little as overlap as possible, since overlap causes the classification error). Because the Kohonen feature maps are the computationally most effective self-organizing nets, they are in fact the best neural net for such problems.

However, it is also possible to use an SRN to teach a representation in the neural filter. A disadvantage of the application of an SRN in the neural filter is the fact that the model implements a infinite order Markov chain by using recurrent fibres. This is just much too sophisticated. If we use a regular BP net with a window, the net does not form a representation as good as the Kohonen feature map. The representation is much more discontinuous.

Moreover, it is hard to structure the input set. In the case of the neural interest map, either the SRN or the Kohonen feature map does well. Both run out of addressing space and both have shown to be pretty good in such clustering problems. An advantage of the Kohonen feature map might be a faster and more stable convergence. Recent simulations of SRN's in IR showed very long training times. However, the cluster boundary determination problem is much harder in Kohonen feature maps than in the SRN. Future Kohonen net types might solve this problem.

If one wants to learn a specific mapping or function approximation, then back-propagation seems to be the best choice. However, one has to realize that most IR problems are clustering problems and not mapping problems, making BP a second choice.

## Future Research

Future research concentrates on different items. First there is the implementation of higher order linguistic structures based on the Data Oriented Parsing (DOP) paradigm. Next, the incorporation of knowledge representation structures implemented by hierarchical feature maps shall be evaluated. Finally, network types that adapt themselves to the underlying probability function are evaluated on their usefulness in information retrieval.

### *Data Oriented Parsing*

In the DOP paradigm, words as well as their syntactical and semantical categorizations are stored in an efficiently implemented data base. Parsing consists of matching (partial) sentences to known sentence structures in the data base. The exact matching algorithm and efficient implementation are main topics in this research. For the moment, high order Markov chains represent the temporal aspects of the sentences and complicated mathematics implement the matching functions. The research carried out here shows that certain neural nets are very well suited to implement such information efficiently. Therefore, a model incorporating sentences and their syntactic categories in information retrieval is begin developed and evaluated at the very moment [Scha, 1990].

### *Hierarchical Feature Maps*

Many researchers have used the concept of hierarchical feature maps to represent relations between concepts [Miikulainen, 1990a,b]. However, the main problem is the connections between the feature maps; how does one define them and how are they incorporated in the learning process [Kangas, 1990], [Samarabunda et al., 1990], [Stotzka et al., 1990], [Tacker et al., 1990], [Ichiki et al., 1991], [Kohonen, 1991].

On the other hand, if the maps are derived (manually or automatically) they provide a great tool for the integration of knowledge structures in information retrieval. These feature maps might be on their own, they can be combined with the automatic derivation of synonym groups, as carried out in [Scholtes, 1991a-c], or they can be incorporated in the DOP simulations.

More on the classic ideas in hierarchical document organization can be found in [Jardine et al., 1971], [Willet, 1979, 1984, 1988]

### *Growing Net Structures*

The most promising and most important future research in neural information retrieval is the evaluation of growing net structures. By automatically deriving the best (clustered) structure for a specific probability distribution, the effects of the neighbourhoods in the neural filter as well as the problems with the cluster boundaries in the neural interest map may be solved in an elegant way [Fritzke, 1991a-c], [Martinetz et al., 1991].

### *Genetic Algorithms in Information Retrieval*

To avoid the very undeterministic character in Neural Net applications (the model either works or not, local optimizations are hard or impossible), an increased interest in genetic IR can be spotted. Here, only some pointers are mentioned. Detailed discussion is outside the scope of this paper [Bennett et al., 1991], [DeJong et al., 1989], [Sharma, 1989], [Siegelman et al., 1991]

## Acknowledgements

The author wishes to thank Remko Scha for many fruitful discussions over the last two and half years. Bob Allen for bringing up the idea of neural information retrieval and Hans Henseler for helping generating the postscript pictures.

Furthermore, Brian Bartell, Bernd Fritzke, Professor Kohonen, Professor Oja and Helge Ritter for the time they took to discuss the issues brought up in this report and to comment on the results.

A final word of thanks remains for all the members of the DIPDOP colloquium group of the University of Amsterdam, who commented on earlier versions of this work.

Parts of this work have been published in [Scholtes, 1991d-f, 1992a-d].

## References

[Allen, 1991]: Allen, R.B. (1991). Knowledge Representation and Information Retrieval with Simple Recurrent Networks. Work. Notes of the AAAI SSS on Connectionist Natural Language Processing. March 26th-28th 1991, Palo Alto, CA., pp. 1-6.

[Barrett, 1989]: Barrett, E. (Editor) (1989). The Society of Text. MIT Press.

[Bein et al., 1988]: Bein, J. and Smolensky, P. (1988). Applications of the Interactive Activation Model to Document Retrieval. Technical Report CU-CS-405-88, University of Colorado, Boulder, CO.

[Belew et al., 1988]: Belew, R.K. and Holland, M.P. (1988). A Computer System Designed to Support the Near-Library User of Information Retrieval. Microcomputers for Information Management, Vol. 5, No. 3, December 1988, pp. 147-167.

[Belew, 1986]: Belew, R.K. (1986). Adaptive Information Retrieval: Machine Learning in Associative Networks. Ph.D. Dissertation, Univ. Michigan, CS Department, Ann Harbor, MI.

[Belew, 1987]: Belew, R.K. (1987). A Connectionist Approach to Conceptual Information Retrieval. Proc. of the First Intl. conf. on AI and Law, pp. 116-126. ACM Press.

[Belew, 1989]: Belew, R.K. (1989). Adaptive Information Retrieval: Using a Connectionist Representation to Retrieve and Learn About Documents. Proc. of the 12th SIGIR-89. June 11th-20th, 1989. Cambridge, MA., pp. 11-20.

[Bennett et al., 1991]: Bennett, K., Ferris, M.C., Ioannidis, Y.E. (1991). A Genetic Algorithm for Database Query Optimization. Proc. of the 4th. Int. Conf. on Genetic Algorithms. UCSD, La Jolla, CA., pp. 400-407.

[Bochereau Laurent et al., 1991]: Bochereau Laurent (1991). Extracting Legal Knowledge by Means of a Multi Layer Neural Net. proc. of the ICAIL,

[Bokhari, 1981]: Bokhari, S.H. (1981). On the Mapping Problem. IEEE Transactions on Computers, Vol. 30, No. 3, pp. 207-214.

[Boyer et al., 1977]: Boyer, R.S., Moore, J.S. (1977). A Fast String Searching Algorithm. Communications of the ACM, Vol. 20, No. 10, pp. 762-772.

[Bozinovic et al., 1982]: Bozinovic, R., Srihari, S.N. (1982). A String Correction Algorithm for Cursive Script Recognition. IEEE Transactions on Pattern Analysis and machine Intelligence, Nov. 1982, pp. 655-663, pp. 236-244.

[Brachman et al., 1988]: Brachman, R.J. and McGuinness, D.L. (1988). Knowledge Representation, Connectionism, and Conceptual Retrieval. SIGIR-88, pp. 161-174.

[Brown et al., 1990]: Brown, P.F., Cocke, J., Della Pietra, S.A., Della Pietra, V.J., Jelinek, F., Lafferty, J.D., Mercer, R.L., and Roossin, P.S. (1990). A Statistical Approach to Machine Translation. Computational Linguistics, Vol. 16, No. 2, pp. 79-85

[Chrisley, 1990]: Chrisley, R.L. (1990). Cognitive Map Construction and Use: A Parallel Distributed Processing Approach. In: Connectionist Models: Proc. of the 1990 Summer School, pp. 287-300. Morgan-Kaufman Publishers.

[Croft et al., 1979]: Croft, W.B., Harper, D.J. (1979). Using Probabilistic Models of Information Retrieval without Relevance Information. Journal of Documentation, Vol. 35, No. 4, pp. 285-295.

[Croft, 1977]: Croft, W.B. (1977). Clustering Large Files of Documents Using the Single Link Method. Journal of the American Soc. for Information Science (JASIS), Vol. 28, No. 6, pp. 341-344.

[Croft, 1980]: Croft, W.B. (1980). A Model of Cluster Searching Based on Classification. Information Systems, Vol. 5, No. 3, pp. 189-195.

[Croft, 1981]: Croft, W.B. (1981). Document Representation in Probabilistic Models in Clustered Files. Journal of the American Soc. for Information Science (JASIS), Vol. 31, No. 6, pp. 451-457.

[Crough, 1986]: Crough, D.B. (1986). The Visual Display of Information in an Information Retrieval Environment. SIGIR-86, pp. 58-67.

[D'Amore et al., 1988]: D'Amore, R. and Mah, C. (1988). N-Grams. PAR Government Systems Corp., McLean, Virginia.

[DeJong et al., 1989]: DeJong, K.A. and Spears, W.M. (1989). Using Genetic Algorithms to Solve NP-Complete Problems. Proc. of the 3rd Int. Conf. on Genetic Algorithms, pp. 124-132.

[Devijver et al., 1982]: Devijver, P.A. and Kittler, J. (1982). Pattern Recognition, A Statistical Approach. Prentice Hall International.

[Doszkocs et al., 1990]: Doszkocs, T.E., Reggia, J. and Lin, X. (1990). Connectionist Models and Information Retrieval. Ann. Review of Information Science and Technology, Vol. 25, pp. 209-260.

[Doyle, 1961]: Doyle, L.B. (1961). Semantic Road Maps for Literature Searchers. Journal of the ACM, Vol. 8, pp. 553-578.

[Duda et al., 1973]: Duda, R.O. and Hart, P.E. (1973). Pattern Classification and Scene Analysis. John Wiley and Sons Inc.

[Eichmann et al., 1991]: Eichmann, D.A. and Srinivas, K. (1991). Neural Network-Based Retrieval from Reuse Repositories. CHI '91 Workshop on Pattern Recognition and Neural Networks in Human-Computer Interaction, April 28th, 1991. New Orleans, LA.

[Eichmann et al., 1992]: Eichmann, D.A. and Srinivas, K. (1992). Neural Network-Based Retrieval from Reuse Repositories. In: Neural Networks and Pattern Recognition in Human Computer Interaction (R. Beale and J. Findlay, Eds.), Ellis Horwood Ltd, WS, UK.

[Elman, 1988]: Elman, J.L. (1988). Finding Structure in Time. Technical Report CRL 8801, Center for the Research of Language, UCSD.

[Forney, 1973]: Forney Jr., G.D. (1973). The Viterbi Algorithm. Proceedings of the IEEE, Vol. LXI, pp. 268-278.

[Frieder et al., 1991]: Frieder, O. and Siegelman, H.T. (1991). On the Allocation of Documents in Multiprocessor Information Retrieval Systems. ACM SIGIR 191, October 1991, Chicago.

[Fritzke, 1991a]: Fritzke, B. (1991). Let it Grow- Self-Organizing Feature Maps With Problem Dependent Cell Structure. Proc. of the ICANN. June 24-28th, 1991. Helsinki, Finland. Vol. 1, pp. 403-408.

[Fritzke, 1991b]: Fritzke, B. (1991). Unsupervised Clustering with growing Cell Structures. Proc. of the IJCNN. July. 8th-12th, 1991. Seattle, WA, Vol. 2, pp. 531-536.

[Fritzke et al., 1991c]: Fritzke, B. and Wilke, P. (1991). FLEXMAP - A Neural Network For The Traveling Salesman Problem With Linear Time and Space Complexity. Submitted to the IJCNN, Singapore, 1991.

[Fu, 1977]: Fu, K.S. (Editor) (1977). Syntactic Pattern Recognition, Applications. Springer-Verlag.

[Furnas et al., 1988]: Furnas, G.W., Deerwester, S., Dumais, S.T., Landauer, T.K., Harshman, R.A. and Streeter, L.A. (1988). Information Retrieval Using a Singular Value Decomposition Model of Latent Sem. Structure. Proc. 11th R&D in IR, pp. 465-480.

[Gersho et al., 1990a]: Gersho, M. and Reiter, R. (1990). Information Retrieval using a Hybrid Multi-Layer Neural Network. Proceedings of the IJCNN, San Diego, June 17-21, 1990, Vol. 2, pp. 111-117.

[Gersho et al., 1990b]: Gersho, M. and Reiter, R. (1990). Information Retrieval using Self-Organizing and Heteroassociative Supervised Neural Network. Proc. of the INNC, Paris, Vol. 1, pp. 361-364.

[Hanson et al., 1974]: Hanson, A.R., Riseman, E.M. (1974). A Contextual Postprocessing System for Error Correction Using Binary n-Grams. IEEE Transactions on Computers, May 1974, pp. 480-493, pp. 141-154.

[Harrison, 1971]: Harrison, M.C. (1971). Implementation of the Substring Test by Hashing. Communications of the ACM, Vol. 14, No. 12, pp. 777-779.

[Hingston et al., 1990]: Hingston, P. and Wilkinson, R. (1990). Document Retrieval Using a Neural Network. Technical Report, Department of Computer Science, RMIT at the University of Melbourne, 1990.

[Honkela et al., 1991]: Honkela, T. and Vepsalainen, A.M. (1991). Interpreting Imprecise Expressions: Experiments with Kohonen's Self-Organizing Maps and Associative Memory. Proc. of the ICANN '91, June 24th-28th, Helsinki, Vol. 1, pp. 897-902.

[Hull et al., 1982]: Hull, J.J., Srihari, S.N. (1982). Experiments in text Recognition with Binary n-Grams and Viterbi Algorithms. IEEE Transactions on Pattern Analysis and Machine Intelligence, Sept. 1982, pp. 520-530, pp. 162-172.

[Ichiki et al., 1991]: Ichiki, H., Hagiwara, M., Nakagawa, N. (1991). Self-Organizing Multi-Layer Semantic Maps. Proc. Of the IJCNN. July. 8th-12th, 1991. Seattle, WA, Vol. 1, pp. 357-360.

[Jagota et al., 1990]: Jagota, A. and Hung, Y.-S. (1990). A Neural Lexicon in a Hopfield-style Network. Proceedings of the IJCNN, Washington, 1990, Vol. 2, pp. 607-610.

[Jagota, 1990a]: Jagota, A. (1990). Applying a Hopfield-style Network to Degraded Printed Text Restoration. Proc. of the 3rd Conference on Neural Networks and PDP, Indiana-Purdue University.

[Jagota, 1990b]: Jagota, A. (1990). Applying a Hopfield-Style Network to Degraded Text Recognition. Proc. of the IJCNN 1990, Washington DC, Vol. 2, pp. 607-610.

[Jagota, 1991]: Jagota, A. (1991). Novelty Detection on a Very Large Number of Memories in a Hopfield-style Network. Proc. of the IJCNN, July 8th-12th, 1991. Seattle, WA, Vol. 2, pp. 905.

[Jardine et al., 1971]: Jardine, N., Rijsbergen van, C.J. (1971). The Use of Hierarchic Clustering in Information Retrieval. Information Storage and Retrieval, Vol. 7, No. 5, pp. 217-240.

[Jelinek et al., 1989]: Jelinek, F., Fusijaki, T., Cocke, J., Black, E. and Nishino, T. (1989). A Probabilistic Parsing Method for Sentence Disambiguation. CMU International Parsing Workshop, 1989, pp. 85-94.

[Jelinek et al., 1991a]: Jelinek, F. and Lafferty, J.D. (1991). Computation of the Probability of Initial Substring Generation by Stochastic Context Free Grammars. IBM, T.J. Watson Research Center, New York.

[Jelinek et al., 1991b]: Jelinek, F. (1991). Self-Organized Language Modelling for Speech Recognition. IBM, T.J. Watson Research Center, New York.

[Jelinek et al., 1991c]: Jelinek, F. (1991). Up from Trigrams: The Struggle for Improved Language Models. IBM, T.J. Watson Research Center, New York.

[Jung et al., 1991]: Jung, G.S. and Raghavan, V.V. (1991). Connectionist Learning in Constructing Thesaurus-Like Knowledge Structure. AAAI Symposium on Text-Based Inteligent Systems, Stanford University, March 1990.

[Kangas, 1990]: Kangas, J. (1990). Time-Delayed Self-Organizing Maps. Proceedings of the IJCNN, San Diego, June 17-21, 1990, Vol. 2, pp. 331-336.

[Kelly, 1991]: Kelly, M. (1991). Self-Organizing Map Training Using Dynamic K-D Trees. Proc. of the ICANN '91, June 24th-28th, Helsinki, Vol. 2, pp. 1041-1044.

[Kimbrell, 1988]: Kimbrell, R.E. (1988). Searching for Text? Send an N-Gram! Byte, May 1988, pp. 297-312.

[Knuth et al., 1977]: Knuth, D.E., Morris, J.H., Pratt, V.R. (1977). Fast Pattern Matching In Strings. SIAM Journal of Computing, Vol. 6, No. 2, pp. 323-350.

[Kohonen, 1982a]: Kohonen, T. (1982). Analysis of a Simple Self-Organizing Process. Biological Cybernetics, Vol. 44, pp. 135-140.

[Kohonen, 1982b]: Kohonen, T. (1982). Clustering, Taxonomy, and Topological Maps of Patterns. Proceedings of the 6th International Conference on Pattern Recognition, pp. 114-128.

[Kohonen, 1982c]: Kohonen, T. (1982). Self-Organized Formation of Topologically Correct Feature Maps. Biological Cybernetics, Vol. 43, pp. 59-69.

[Kohonen, 1984]: Kohonen, T. (1984). Self-Organization and Associative Memory. Springer-Verlag.

[Kohonen, 1988]: Kohonen, T. (1988). An Introduction to Neural Computing. Neural Networks, Vol. 1, nr. 1, pp. 3-16.

[Kohonen, 1990a]: Kohonen, T. (1990). Some Practical Aspects of the Self-Organizing Maps. Proceedings of the IJCNN, Washington, 1990, Vol. 2, pp. 253-256.

[Kohonen, 1990b]: Kohonen, T. (1990). The Self-Organizing Map. Proc. of the IEEE, Vol. 78, pp. 1464-1480.

[Kohonen, 1991]: Kohonen, T. (1991). The Hypermap Architecture. Proc. of the ICANN '91, June 24th-28th, Helsinki, Vol. 2, pp. 1357-1362.

[Koikkalainen et al., 1990]: Koikkalainen, P. and Oja, E. (1990). Self-Organizing Hierarchical Feature Maps. Proceedings of the IJCNN, San Diego, June 17-21, 1990, Vol. 2, pp. 279-284.

[Kwok, 1989]: Kwok, K.L. (1989). A Neural Network for Probabilistic Information Retrieval. Proc. of the 12th SIGIR. June 11th-20th, 1989. Cambridge, MA., pp. 21-30.

[Kwok, 1990]: Kwok, K.L. (1990). Application of Neural Network to Information Retrieval. Proc. of the IJCNN, Januari 15-19, 1990. Washington DC, Vol. 2, pp. 623-626.

[Lakoff, 1988]: Lakoff, G. (1988). A Suggestion for a Linguists with Connectionist Foundations. Proceedings of the Connectionist Models Summer School, Carnegie Mellon University, pp. 301-314.

[Lancaster, 1979]: Lancaster, F.W. (1979). Information Retrieval Systems: Characteristics, Testing and Evaluation (Second Edition). John Wiley and Sons, New York.

[Larson, 1988]: Larson, P.A. (1988). Dynamic Hash Tables. Communications of the ACM, Vol. 31, nr. 4, pp. 446-457.

[Lelu, 1989]: Lelu, A. (1989). Local Component Analysis: A Neural Model for Information Retrieval. Proc. of the IJCNN, Washington DC, Vol. 2, pp. 43-47.

[Lelu, 1991]: Lelu, A. (1991). From Data Analysis to Neural Networks: New Prospects for Efficient Browsing through Databases. Journal of Information Science, Vol. 17, pp. 1-12. Elsevier.

[Lin et al., 1991]: Lin, X., Soergel, D. and Marchionini, G. (1991). A Self-Organizing Semantic Map for Information Retrieval.

[Martinetz et al., 1991]: Martinetz, T. and Schulten, K. (1991). A "Neural-Gas" Network Learns Topologies. Proc. of the ICANN '91, June 24th-28th, Helsinki, Vol. 1, pp. 397-402.

[McIllroy, 1982]: McIllroy, M.D. (1982). Development of a Spelling List. IEEE Transactions on Communications, Vol. COM-30, pp. 91-99.

[Miikkulainen, 1990a]: Miikkulainen, R. (1990). A Distributed Feature Map Model of the Lexicon. Proceedings of the 12th Ann. Conf. of the Cog. Sci. Soc., Cambridge, MA, July 25-28, 1990, pp. 447-454.

[Miikkulainen, 1990b]: Miikkulainen, R. (1990). Script Recognition with Hierarchical Feature Maps. Connection Science, Vol. 2, No. 1&2, pp. 83-102.

[Mitzman et al., 1990]: Mitzman, D. and Giovannini, R. (1990). ActivityNets: A Neural Classifier of Natural Language Descriptions of Economic Activities. Proc. of the Int. Works. on Neural Nets for Stat. and Economic Data, Dublin, December 10-11, 1991.

[Mozer, 1984]: Mozer, M.C. (1984). Inductive Information Retrieval Using Parallel Distributed Computation. TR-ICS 8406, UCSD, La Jolla, CA.

[Mozer, 1991]: Mozer, M.C. (1991). Discovering Discrete Distributed Representations. Advances in NIPS 3 (R.P. Lippmann, J.E. Moody and D.S. Touretzky, Eds.), pp. 627-634. Morgan Kaufmann.

[Neuhoff, 1975]: Neuhoff, D.L. (1975). The Viterbi Algorithm As an Aid in Text Recognition. IEEE Transactions on Information Theory, Vol. IT-XXI, pp. 222-226.

[Oddy et al., 1991]: Oddy, R.N. and Balakrishnan, B. (1991). PThomas: an Adaptive Information Retrieval System on the Connection Machine. Information Processing and Management,

[Oja et al., 1988]: Oja, E. and Kohonen, T. (1988). The Subspace Learning Algorithm as a Formalism for Pattern Recognition and Neural Networks. Proceedings of the 2nd International Conference on Neural Networks, San Diego, Vol. 1, pp. 277-284.

[Oja et al., 1991]: Oja, E., Ogawa, H. and Wangviwattana, J. (1991). Learning in Nonlinear Constrained Hebbian Networks. Proc. of the ICANN '91, June 24th-28th, Helsinki, Vol. 1, pp. 385-390.

[Palakal et al., 1991]: Palakal, M. and Thai, X. (1991). Computational Model of a Biologically Plausible Cognitive map. Proc. of the IJCNN. July. 8th-12th, 1991. Seattle, WA, Vol. 2, pp. A-885.

[Personnaz et al., 1986]: Personnaz, L., Guyon, I. and Dreyfus, G. (1986). Neural Network Design for Efficient Information Retrieval. In: Disordered Systems and Biological Organization (E. Bienenstock et al., Eds.), Springer-Verlag.

[Pogue et al., 1987]: Pogue, C.A. and Willett, P. (1987). Use of Text Signatures for Document Retrieval in a Highly Parallel Environment. Parallel Computing, Vol. 4, pp. 259-268.

[Rapp, 1991]: Rapp, R., Wettler, M. (1991). A Connectionist Simulation of Human Word Associations. Proc. Of the IJCNN. July. 8th-12th, 1991. Seattle, WA, Vol. 2, pp. A-946.

[Regier, 1988]: Regier, T. (1988). Recognizing Image-Schemes Using Programmable Networks. Proceedings of the Connectionist Models Summer School, CMU.

[Rijsbergen, 1979]: Rijsbergen, C.J. van (1979). Information Retrieval. Butterworths, London.

[Ritter et al., 1989a]: Ritter, H. and Schulten, K. (1989). Convergency Properties of Kohonen's Topology Conserving Maps: Fluctuations, Stability and Dimension Selection. Biological Cybernetics, Vol. 60, pp. 59-71.

[Ritter et al., 1989b]: Ritter, H. and Kohonen, T. (1989). Self-Organizing Semantical Maps. Biological Cybernetics, Vol. 61, pp. 241-254.

[Ritter et al., 1990]: Ritter, H. and Kohonen, T. (1990). Learning "Semantotopic Maps" from Context. Proceedings of the IJCNN, Washington, 1990, Vol. 1, pp. 23-26.

[Ritter, 1991]: Ritter, H. (1991). Learning with the Self-Organizing Map. Proc. of the ICANN '91, June 24th-28th, Helsinki, Vol. 1, pp. 379-384.

[Rocchio, 1971]: Rocchio, J.J. (1971). Relevance Feedback in Information Retrieval. In: The Smart Retrieval System (G. Salton, Editor), Prentice Hall, New Jersey.

[Rose et al., 1989a]: Rose, D.E. and Belew, R.K. (1989). A case for Symbolic/Sub-Symbolic Hybrids. Proc. Of the 11th Ann. Conf. of the Cog. Sci. Soc., August. 16th-19th, 1989. Ann Arbor, pp. 844-851.

[Rose et al., 1989b]: Rose, D.E. and Belew, R.K. (1989). A Case for Symbolic/Sub-Symbolic Hybrids. Proc. of the Cogn. Science Society, pp. 844-851.

[Rose et al., 1991]: Rose, D.E. and Belew, R.K. (1991). A Connectionist and Symbolic Hybrid for Improving Legal Research. Int. Journal of Man-Machine Studies, July 1991.

[Rose, 1990]: Rose, D.E. (1990). Appropriate Uses of Hybrid Systems. In: Connectionist Models: Proc. of the 1990 Summer School, pp. 277-286. Morgan-Kaufman Publishers.

[Rose, 1991]: Rose, D.E. (1991). A Symbolic and Connectionist Approach to Legal Information Retrieval. Ph.D. Dissertation, UCSD, La Jolla, CA.

[Salton et al., 1968]: Salton, G., Wong, A., Yang, C.S. (1968). A Vector Space Model for Automatic Indexing. Communications of the ACM, Vol. 18, No. 11, pp. 613-620.

[Salton et al., 1973]: Salton, G., Yang, C.S. (1973). On the Specification of Term Values in Automatic Indexing. Journal of Documentation, Vol. 29, no. 4, pp. 351-372.

[Salton et al., 1983]: Salton, G., Fox, E.A., Wu, H. (1983). Extended Boolean Information Retrieval. Communications of the ACM, Vol. 26, No. 11, pp. 189-195.

[Salton et al., 1988]: Salton, G. and Buckley, C. (1988). Parallel Text Search Methods. Communications of the ACM, Vol. 31, No. 2, pp. 202-215.

[Salton, 1968]: Salton, G. (1968). Automatic Information Organization and Retrieval. McGraw-Hill Book Co., New York.

[Salton, 1971]: Salton, G. (Editor) (1971). The Smart Retrieval System. Prentice Hall, New Jersey.

[Salton, 1972]: Salton, G. (1972). Experiments in Automatic Thesaurus Construction for information Retrieval. Information Processing 71, Vol. 71, pp. 115-123.

[Salton, 1980]: Salton, G. (1980). Automatic Information Retrieval. Computer, Vol. 13, No. 9, pp. 41-57.

[Salton, 1986]: Salton, G. (1986). Another Look at Automatic Text Retrieval. Communications of the ACM, Vol. 29, No. 7, pp. 648-656.

[Salton, 1989]: Salton, G. (1989). Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer. Addison-Wesley Publishing Corporation.

[Samarabandu, et al., 1990]: Samarabandu, J.K. and Jakubowics, O.G. (1990). Principles of Sequential Feature Maps in Multi-level Problems. Proceedings of the IJCNN, Washington, 1990, Vol. 2, pp. 683-686.

[Sammon, 1969]: Sammon, J.W. (1969). A Nonlinear Mapping for Data Structure Analysis. IEEE Transactions on Computers, Vol. 18, pp. 401-409.

[Scha, 1990]: Scha, R. (1990). Taaltheorie en Taaltechnologie; Competence en Performence (In Dutch). In: Computertoepassingen in de Neerlandistiek (K. de Kort & G.L.J. Leerdam, Eds.), pp. 99-109. Almere: LVVN.

[Scholtes, 1991a]: Scholtes, J.C. (1991). Recurrent Kohonen Self-Organization in Natural Language Processing. Proceedings of the ICANN, pp. 1751-1754. June 26th-29th 1991, Helsinki Finland.

[Scholtes, 1991b]: Scholtes, J.C. (1991). Unsupervised Context Learning in Natural Language Processing. Proceedings of the IJCNN, Vol. 1, pp. 107-112. July 8th-12th 1991, Seattle.

[Scholtes, 1991c]: Scholtes, J.C. (1991). Kohonen Feature Maps in Natural Language Processing. Technical Report, March 1991, Department of Computational Linguistics, University of Amsterdam.

[Scholtes, 1991d]: Scholtes, J.C. (1991). Filtering the Pravda with a Self-Organizing Neural Net. Worknotes of the Bellcore Workshop on High Performance Information Filtering, November 5th-7th 1991, Chester, NJ.

[Scholtes, 1991e]: Scholtes, J.C. (1991). Unsupervised Learning and the Information Retrieval Problem. Proceedings of the IJCNN, November 18-21, Singapore.,

[Scholtes, 1991f]: Scholtes, J.C. (1991). Kohonen Feature Maps and Full-Text Data Bases: A Case Study of the 1987 Pravda. Submitted to Informatiewetenschappen, 1991. December 18th. Nijmegen, The Netherlands,

[Scholtes, 1992a]: Scholtes, J.C. (1992). Neural Nets and Full Text Information Filtering. Submitted to the ACNN '92. February 3-5, 1992. Canberra, Australia,

[Scholtes, 1992b]: Scholtes, J.C. (1992). Various Applications of Self-Organizing Neural Nets in Information Retrieval. Submitted to the 3rd Applied NLP ACL, march 1st-3rd, Trento, Italy,

[Scholtes, 1992c]: Scholtes, J.C. (1992). Filtering the Pravda with a Self-Organizing Neural Net. Submitted to the Symposium on Document Analysis and Information Retrieval. March 16th-18th 1992, Las Vegas, NV.

[Scholtes, 1992d]: Scholtes, J.C. (1992). Neural Nets versus Statistics in Information Retrieval. A Case Study of the 1987 Pravda. Proceedings of the SPIE Conference on Applications of Artificial Neural Networks III. 20-24 April 1992. Orlando, Florida.

[Sharkey et al., 1989]: Sharkey, A.J.C., Sharkey, N.E. (1989). Lexical Processing and the mechanism of Context Effects in text Comprehension. Proc. Of the 11th Ann. Conf. of the Cog. Sci. Soc., August. 16th-19th, 1989. Ann Arbor, pp. 466-473.

[Sharma, 1989]: Sharma, R. (1989). A Generic Machine for Parallel Information Retrieval. Information Processing and Management, Vol. 25, no. 3, pp. 223-235.

[Shinghal et al., 1979a]: Shinghal, R., Toussaint, G.T. (1979). A Bottom-Up and Top-Down Approach to Using Context in text Recognition. International Journal of Man-Machine Studies, 1979, pp. 201-212, pp. 174-181.

[Shinghal et al., 1979b]: Shinghal, R, Toussaint, G.T. (1979). Experiments in Text Recognition with the modified Viterbi Algorithm. IEEE Transactions on Pattern analysis and Machine Intelligence, April 1979, pp. 184-193, pp. 131-140.

[Siedlecki et al., 1988]: Siedlecki, W., Siedlecki, K., and Sklansky, J. (1988). An Overview of Mapping Techniques for Exploratory Pattern Analysis. Pattern Recognition, Vol. 21, No. 5, pp. 411-429.

[Siegelman et al., 1991]: Siegelman, H.T. and Frieder, O. (1991). The Allocation of Documents in Multiprocessor Information Retrieval Systems: An Application of Genetic Algorithms. IEEE System, Man and Cybernetics, Ocotber 1991.

[Small et al., 1974]: Small, H. and Griffith, B.C. (1974). The Structure of Scientific Literature I: Identifying and Graphing Specialities. Social Studies, Vol. 4, pp. 17-40.

[Sparck Jones, 1971]: Sparck Jones, K. (1971). Automatic Keyword Classification for Information Retrieval. Butterworth, London.

[Srihari et al., 1983]: Srihari, S.N., Hull, J.J. (1983). Integrating Diverse Knowledge Sources in Text Recognition. ACM Transactions on Office Information systems, Jan. 1983, pp. 68-87, pp. 216-235.

[Srihari, 1985]: Shrihari, S.N. (1985). Computer Text Recognition and Error Correction: a Tutorial. IEEE Computer Society.

[Stanfill et al., 1986]: Stanfill, C.W. and Kahle, B. (1986). Parallel Free-Text Search on the Connection Machine System. Communications of the ACM, Vol. 29, pp. 1229-1239.

[Stanfill et al., 1989]: Stanfill, C.W., Thau, R. and Waltz, D.L. (1989). A Parallel Indexed Algorithm for Information Retrieval. Proc. of the 12th Intl. Conf. on Research and Development in Information Retrieval (SIGIR-89), June 26th-28th 1989.

[Stotzka et al., 1990]: Stotzka, R. and Manner, R. (1990). Self-Organization in a Linear Multi-Layered Feedforward Neural Network. Proceedings of the IJCNN, Washington, 1990, Vol. 1, pp. 126-129.

[Tenopir et al., 1990]: Tenopir, C. and Ro, J.S. (1990). Full Text Databases. Greenwood Publishing Group, Inc.

[Tenopir, 1984]: Tenopir, C. (1984). Full-Text Databases. Annual Review of Information Science and Technology (Ed. William, N.), Vol. 19, pp. 215-246.

[Thacker et al., 1990]: Thacker, N.A. and Mayhew, J.E.W. (1990). Designing a Layered Network for Context Sensitive Pattern Classification. Neural Networks, Vol. 3, No. 3, pp. 291-299.

[Turtle et al., 1991]: Turtle, H. and Croft, W.B. (1991). Evaluation of Inference Network-based Retrieval Models. ACM Transactions on Information Systems, (In Press).

[Van Opdorp et al., 1991]: Van Opdorp, G.J. (1991). Networks at Work: A Connectionist Approach to Non-Deductive Legal Reasoning. Proc.of the ICAIL,

[Voorhees, 1985]: Voorhees, E.M. (1985). The Cluster Hypothesis Revisited. Proc. of the 8th. Ann. Int. ACM SIGIR Conf. on Research and Development in Information Retrieval, June '85, pp. 188-196.

[Waltz, 1990]: Waltz, D.L. (1990). Memory-based Reasoning. In: Natural and Artificial Parallel Computation (M.A. Arbib and A. Robinson, Eds.), pp. 251-276. MIT Press.

[Wermter, 1991]: Wermter, S. (1991). Learning to Classify Neural Language Titles in a Recurrent Connectionist Model. Proc. of the ICANN '91, June 24th-28th, Helsinki, Vol. 2, pp. 1715-1718.

[Wettler et al., 1989]: Wettler, M., Rapp, R. (1989). A Connectionist System to Simulate Lexical Decisions in Information Retrieval. In: Connectionism in Perspective (R. Pfeiffer et al., Eds.), pp. 463-469. North-Holland.

[Wettler et al., 1990]: Wettler, M., Rapp, R. (1990). Parallel Associative Processes in Information Retrieval. In: Parallel Processing in Neural Systems and Computers (R. Eckmiller, G. Hartmann and G. Hauske, Eds), pp. 509-512. Elsevier.

[Weyer, 1989]: Weyer, S. (1989). Question for the Dao: DowQuest and Intelligent Text Retrieval. Online, Vol. 13, No. 5, pp. 39-48.

[Wilbert, 1991]: Wilbert, R. (1991). Assoziative Begriffsrepresentation in Neuronalen Netzen. Nachrichten fuer Dokumentation, Vol. 42, No. 3, pp. 205-211.

[Wilkinson et al., 1991]: Wilkinson, R. and Hingston, P. (1991). Using the Cosine Measure in a Neural Network for Document Retrieval. Proc. of the SIGIR '91 on Research and Development in Information Retrieval, October 13th-16th, 1991. Chicago, MI.

[Willett, 1979]: Willett, P. (1984). Document Retrieval Experiments Using Indexing Vocabularies od Varying Size, II. Hashing, Truncation, Digram and Trigram Encoding of Index Terms. Journal of Documentation, Vol. 35, No. 4, pp. 215-246.

[Willett, 1984]: Willett, P. (1984). A Note on the Use of Nearest Neighbors for Implementing Single Linkage Document Classification. Journal of the ASIS, Vol. 35, No. 3, pp. 149-152.

[Willett, 1988]: Willett, P. (1988). Recent Trends in Hierarchical Document Clustering: A Critical Review. Information Processing and Management, Vol. 24, No. 5, pp. 577-597.

# The ITLI Prepublication Series

LP-90-13 Zhisheng Huang — Logics for Belief Dependence
LP-90-14 Jeroen Groenendijk, Martin Stokhof — Two Theories of Dynamic Semantics
LP-90-15 Maarten de Rijke — The Modal Logic of Inequality
LP-90-16 Zhisheng Huang, Karen Kwast — Awareness, Negation and Logical Omniscience
LP-90-17 Paul Dekker — Existential Disclosure, Implicit Arguments in Dynamic Semantics

*Mathematical Logic and Foundations*

ML-90-01 Harold Schellinx — Isomorphisms and Non-Isomorphisms of Graph Models
ML-90-02 Jaap van Oosten — A Semantical Proof of De Jongh's Theorem
ML-90-03 Yde Venema — Relational Games
ML-90-04 Maarten de Rijke — Unary Interpretability Logic
ML-90-05 Domenico Zambella — Sequences with Simple Initial Segments
ML-90-06 Jaap van Oosten — Extension of Lifschitz' Realizability to Higher Order Arithmetic, and a Solution to a Problem of F. Richman
ML-90-07 Maarten de Rijke — A Note on the Interpretability Logic of Finitely Axiomatized Theories
ML-90-08 Harold Schellinx — Some Syntactical Observations on Linear Logic
ML-90-09 Dick de Jongh, Duccio Pianigiani — Solution of a Problem of David Guaspari
ML-90-10 Michiel van Lambalgen — Randomness in Set Theory
ML-90-11 Paul C. Gilmore — The Consistency of an Extended NaDSet

*Computation and Complexity Theory*

CT-90-01 John Tromp, Peter van Emde Boas — Associative Storage Modification Machines
CT-90-02 Sieger van Denneheuvel, Gerard R. Renardel de Lavalette — A Normal Form for PCSJ Expressions
CT-90-03 Ricard Gavaldà, Leen Torenvliet, Osamu Watanabe, José L. Balcázar — Generalized Kolmogorov Complexity in Relativized Separations
CT-90-04 Harry Buhrman, Edith Spaan, Leen Torenvliet — Bounded Reductions
CT-90-05 Sieger van Denneheuvel, Karen Kwast — Efficient Normalization of Database and Constraint Expressions
CT-90-06 Michiel Smid, Peter van Emde Boas — Dynamic Data Structures on Multiple Storage Media, a Tutorial
CT-90-07 Kees Doets — Greatest Fixed Points of Logic Programs
CT-90-08 Fred de Geus, Ernest Rotterdam, Sieger van Denneheuvel, Peter van Emde Boas — Physiological Modelling using RL
CT-90-09 Roel de Vrijer — Unique Normal Forms for Combinatory Logic with Parallel Conditional, a case study in conditional rewriting

*Other Prepublications*

X-90-01 A.S. Troelstra — Remarks on Intuitionism and the Philosophy of Mathematics, Revised Version
X-90-02 Maarten de Rijke — Some Chapters on Interpretability Logic
X-90-03 L.D. Beklemishev — On the Complexity of Arithmetical Interpretations of Modal Formulae
X-90-04 — Annual Report 1989
X-90-05 Valentin Shehtman — Derived Sets in Euclidean Spaces and Modal Logic
X-90-06 Valentin Goranko, Solomon Passy — Using the Universal Modality: Gains and Questions
X-90-07 V.Yu. Shavrukov — The Lindenbaum Fixed Point Algebra is Undecidable
X-90-08 L.D. Beklemishev — Provability Logics for Natural Turing Progressions of Arithmetical Theories
X-90-09 V.Yu. Shavrukov — On Rosser's Provability Predicate
X-90-10 Sieger van Denneheuvel, Peter van Emde Boas — An Overview of the Rule Language RL/1
X-90-11 Alessandra Carbone — Provable Fixed points in $I\Delta_0+\Omega_1$, revised version
X-90-12 Maarten de Rijke — Bi-Unary Interpretability Logic
X-90-13 K.N. Ignatiev — Dzhaparidze's Polymodal Logic: Arithmetical Completeness, Fixed Point Property, Craig's Property
X-90-14 L.A. Chagrova — Undecidable Problems in Correspondence Theory
X-90-15 A.S. Troelstra — Lectures on Linear Logic

**1991**

*Logic, Semantics and Philosophy of Langauge*

LP-91-01 Wiebe van der Hoek, Maarten de Rijke — Generalized Quantifiers and Modal Logic
LP-91-02 Frank Veltman — Defaults in Update Semantics
LP-91-03 Willem Groeneveld — Dynamic Semantics and Circular Propositions
LP-91-04 Makoto Kanazawa — The Lambek Calculus enriched with additional Connectives
LP-91-05 Zhisheng Huang, Peter van Emde Boas — The Schoenmakers Paradox: Its Solution in a Belief Dependence Framework
LP-91-06 Zhisheng Huang, Peter van Emde Boas — Belief Dependence, Revision and Persistence

*Mathematical Logic and Foundations*

ML-91-01 Yde Venema — Cylindric Modal Logic
ML-91-02 Alessandro Berarducci, Rineke Verbrugge — On the Metamathematics of Weak Theories
ML-91-03 Domenico Zambella — On the Proofs of Arithmetical Completeness for Interpretability Logic
ML-91-04 Raymond Hoofman, Harold Schellinx — Collapsing Graph Models by Preorders
ML-91-05 A.S. Troelstra — History of Constructivism in the Twentieth Century
ML-91-06 Inge Bethke — Finite Type Structures within Combinatory Algebras
ML-91-07 Yde Venema — Modal Derivation Rules
ML-91-08 Inge Bethke — Going Stable in Graph Models
ML-91-09 V.Yu. Shavrukov — A Note on the Diagonalizable Algebras of PA and ZF
ML-91-10 Maarten de Rijke, Yde Venema — Sahlqvist's Theorem for Boolean Algebras with Operators

*Computation and Complexity Theory*

CT-91-01 Ming Li, Paul M.B. Vitányi — Kolmogorov Complexity Arguments in Combinatorics
CT-91-02 Ming Li, John Tromp, Paul M.B. Vitányi — How to Share Concurrent Wait-Free Variables
CT-91-03 Ming Li, Paul M.B. Vitányi — Average Case Complexity under the Universal Distribution Equals Worst Case Complexity
CT-91-04 Sieger van Denneheuvel, Karen Kwast — Weak Equivalence
CT-91-05 Sieger van Denneheuvel, Karen Kwast — Weak Equivalence for Constraint Sets
CT-91-06 Edith Spaan — Census Techniques on Relativized Space Classes
CT-91-07 Karen L. Kwast — The Incomplete Database
CT-91-08 Kees Doets — Levationis Laus
CT-91-09 Ming Li, Paul M.B. Vitányi — Combinatorial Properties of Finite Sequences with high Kolmogorov Complexity
CT-91-10 John Tromp, Paul Vitányi — A Randomized Algorithm for Two-Process Wait-Free Test-and-Set
CT-91-11 Lane A. Hemachandra, Edith Spaan — Quasi-Injective Reductions

*Computational Linguistics*

CL-91-01 J.C. Scholtes — Kohonen Feature Maps in Natural Language Processing
CL-91-02 J.C. Scholtes — Neural Nets and their Relevance for Information Retrieval

*Other Prepublications*

X-91-01 Alexander Chagrov, Michael Zakharyaschev — The Disjunction Property of Intermediate Propositional Logics
X-91-02 Alexander Chagrov, Michael Zakharyaschev — On the Undecidability of the Disjunction Property of Intermediate Propositional Logics
X-91-03 V. Yu. Shavrukov — Subalgebras of Diagonalizable Algebras of Theories containing Arithmetic
X-91-04 K.N. Ignatiev — Partial Conservativity and Modal Logics
X-91-05 Johan van Benthem — Temporal Logic
X-91-06 — Annual Report 1990
X-91-07 A.S. Troelstra — Lectures on Linear Logic, Errata and Supplement
X-91-08 Giorgie Dzhaparidze — Logic of Tolerance
X-91-09 L.D. Beklemishev — On Bimodal Provability Logics for $\Pi_1$-axiomatized Extensions of Arithmetical Theories
X-91-10 Michiel van Lambalgen — Independence, Randomness and the Axiom of Choice
X-91-11 Michael Zakharyaschev — Canonical Formulas for K4. Part I: Basic Results
X-91-12 Herman Hendriks — Flexibele Categoriale Syntaxis en Semantiek: de proefschriften van Frans Zwarts en Michael Moortgat
X-91-13 Max I. Kanovich — The Multiplicative Fragment of Linear Logic is NP-Complete
X-91-14 Max I. Kanovich — The Horn Fragment of Linear Logic is NP-Complete
X-91-15 V. Yu. Shavrukov — Subalgebras of Diagonalizable Algebras of Theories containing Arithmetic, revised version

# The ITLI Prepublication Series

LP-90-13 Zhisheng Huang — Logics for Belief Dependence
LP-90-14 Jeroen Groenendijk, Martin Stokhof — Two Theories of Dynamic Semantics
LP-90-15 Maarten de Rijke — The Modal Logic of Inequality
LP-90-16 Zhisheng Huang, Karen Kwast — Awareness, Negation and Logical Omniscience
LP-90-17 Paul Dekker — Existential Disclosure, Implicit Arguments in Dynamic Semantics

*Mathematical Logic and Foundations*

ML-90-01 Harold Schellinx — Isomorphisms and Non-Isomorphisms of Graph Models
ML-90-02 Jaap van Oosten — A Semantical Proof of De Jongh's Theorem
ML-90-03 Yde Venema — Relational Games
ML-90-04 Maarten de Rijke — Unary Interpretability Logic
ML-90-05 Domenico Zambella — Sequences with Simple Initial Segments
ML-90-06 Jaap van Oosten — Extension of Lifschitz' Realizability to Higher Order Arithmetic, and a Solution to a Problem of F. Richman
ML-90-07 Maarten de Rijke — A Note on the Interpretability Logic of Finitely Axiomatized Theories
ML-90-08 Harold Schellinx — Some Syntactical Observations on Linear Logic
ML-90-09 Dick de Jongh, Duccio Pianigiani — Solution of a Problem of David Guaspari
ML-90-10 Michiel van Lambalgen — Randomness in Set Theory
ML-90-11 Paul C. Gilmore — The Consistency of an Extended NaDSet

*Computation and Complexity Theory*

CT-90-01 John Tromp, Peter van Emde Boas — Associative Storage Modification Machines
CT-90-02 Sieger van Denneheuvel, Gerard R. Renardel de Lavalette — A Normal Form for PCSJ Expressions
CT-90-03 Ricard Gavaldà, Leen Torenvliet, Osamu Watanabe, José L. Balcázar — Generalized Kolmogorov Complexity in Relativized Separations
CT-90-04 Harry Buhrman, Edith Spaan, Leen Torenvliet — Bounded Reductions
CT-90-05 Sieger van Denneheuvel, Karen Kwast — Efficient Normalization of Database and Constraint Expressions
CT-90-06 Michiel Smid, Peter van Emde Boas — Dynamic Data Structures on Multiple Storage Media, a Tutorial
CT-90-07 Kees Doets — Greatest Fixed Points of Logic Programs
CT-90-08 Fred de Geus, Ernest Rotterdam, Sieger van Denneheuvel, Peter van Emde Boas — Physiological Modelling using RL
CT-90-09 Roel de Vrijer — Unique Normal Forms for Combinatory Logic with Parallel Conditional, a case study in conditional rewriting

*Other Prepublications*

X-90-01 A.S. Troelstra — Remarks on Intuitionism and the Philosophy of Mathematics, Revised Version
X-90-02 Maarten de Rijke — Some Chapters on Interpretability Logic
X-90-03 L.D. Beklemishev — On the Complexity of Arithmetical Interpretations of Modal Formulae
X-90-04 — Annual Report 1989
X-90-05 Valentin Shehtman — Derived Sets in Euclidean Spaces and Modal Logic
X-90-06 Valentin Goranko, Solomon Passy — Using the Universal Modality: Gains and Questions
X-90-07 V.Yu. Shavrukov — The Lindenbaum Fixed Point Algebra is Undecidable
X-90-08 L.D. Beklemishev — Provability Logics for Natural Turing Progressions of Arithmetical Theories
X-90-09 V.Yu. Shavrukov — On Rosser's Provability Predicate
X-90-10 Sieger van Denneheuvel, Peter van Emde Boas — An Overview of the Rule Language RL/1
X-90-11 Alessandra Carbone — Provable Fixed points in $I\Delta_0+\Omega_1$, revised version
X-90-12 Maarten de Rijke — Bi-Unary Interpretability Logic
X-90-13 K.N. Ignatiev — Dzhaparidze's Polymodal Logic: Arithmetical Completeness, Fixed Point Property, Craig's Property
X-90-14 L.A. Chagrova — Undecidable Problems in Correspondence Theory
X-90-15 A.S. Troelstra — Lectures on Linear Logic

**1991**

*Logic, Semantics and Philosophy of Langauge*

LP-91-01 Wiebe van der Hoek, Maarten de Rijke — Generalized Quantifiers and Modal Logic
LP-91-02 Frank Veltman — Defaults in Update Semantics
LP-91-03 Willem Groeneveld — Dynamic Semantics and Circular Propositions
LP-91-04 Makoto Kanazawa — The Lambek Calculus enriched with additional Connectives
LP-91-05 Zhisheng Huang, Peter van Emde Boas — The Schoenmakers Paradox: Its Solution in a Belief Dependence Framework
LP-91-06 Zhisheng Huang, Peter van Emde Boas — Belief Dependence, Revision and Persistence
LP-91-07 Henk Verkuyl, Jaap van der Does — The Semantics of Plural Noun Phrases
LP-91-08 Víctor Sánchez Valencia — Categorial Grammar and Natural Reasoning
LP-91-09 Arthur Nieuwendijk — Semantics and Comparative Logic

*Mathematical Logic and Foundations*

ML-91-01 Yde Venema — Cylindric Modal Logic
ML-91-02 Alessandro Berarducci, Rineke Verbrugge — On the Metamathematics of Weak Theories
ML-91-03 Domenico Zambella — On the Proofs of Arithmetical Completeness for Interpretability Logic
ML-91-04 Raymond Hoofman, Harold Schellinx — Collapsing Graph Models by Preorders
ML-91-05 A.S. Troelstra — History of Constructivism in the Twentieth Century
ML-91-06 Inge Bethke — Finite Type Structures within Combinatory Algebras
ML-91-07 Yde Venema — Modal Derivation Rules
ML-91-08 Inge Bethke — Going Stable in Graph Models
ML-91-09 V.Yu. Shavrukov — A Note on the Diagonalizable Algebras of PA and ZF
ML-91-10 Maarten de Rijke, Yde Venema — Sahlqvist's Theorem for Boolean Algebras with Operators
ML-91-11 Rineke Verbrugge — Feasible Interpretability

*Computation and Complexity Theory*

CT-91-01 Ming Li, Paul M.B. Vitányi — Kolmogorov Complexity Arguments in Combinatorics
CT-91-02 Ming Li, John Tromp, Paul M.B. Vitányi — How to Share Concurrent Wait-Free Variables
CT-91-03 Ming Li, Paul M.B. Vitányi — Average Case Complexity under the Universal Distribution Equals Worst Case Complexity
CT-91-04 Sieger van Denneheuvel, Karen Kwast — Weak Equivalence
CT-91-05 Sieger van Denneheuvel, Karen Kwast — Weak Equivalence for Constraint Sets
CT-91-06 Edith Spaan — Census Techniques on Relativized Space Classes
CT-91-07 Karen L. Kwast — The Incomplete Database
CT-91-08 Kees Doets — Levationis Laus
CT-91-09 Ming Li, Paul M.B. Vitányi — Combinatorial Properties of Finite Sequences with high Kolmogorov Complexity
CT-91-10 John Tromp, Paul Vitányi — A Randomized Algorithm for Two-Process Wait-Free Test-and-Set
CT-91-11 Lane A. Hemachandra, Edith Spaan — Quasi-Injective Reductions

*Computational Linguistics*

CL-91-01 J.C. Scholtes — Kohonen Feature Maps in Natural Language Processing
CL-91-02 J.C. Scholtes — Neural Nets and their Relevance for Information Retrieval
CL-91-03 Hub Prüst, Remko Scha, Martin van den Berg — A Formal Discourse Grammar tackling Verb Phrase Anaphora

*Other Prepublications*

X-91-01 Alexander Chagrov, Michael Zakharyaschev — The Disjunction Property of Intermediate Propositional Logics
X-91-02 Alexander Chagrov, Michael Zakharyaschev — On the Undecidability of the Disjunction Property of Intermediate Propositional Logics
X-91-03 V. Yu. Shavrukov — Subalgebras of Diagonalizable Algebras of Theories containing Arithmetic
X-91-04 K.N. Ignatiev — Partial Conservativity and Modal Logics
X-91-05 Johan van Benthem — Temporal Logic
X-91-06 — Annual Report 1990
X-91-07 A.S. Troelstra — Lectures on Linear Logic, Errata and Supplement
X-91-08 Giorgie Dzhaparidze — Logic of Tolerance
X-91-09 L.D. Beklemishev — On Bimodal Provability Logics for $\Pi_1$-axiomatized Extensions of Arithmetical Theories
X-91-10 Michiel van Lambalgen — Independence, Randomness and the Axiom of Choice
X-91-11 Michael Zakharyaschev — Canonical Formulas for K4. Part I: Basic Results
X-91-12 Herman Hendriks — Flexibele Categoriale Syntaxis en Semantiek: de proefschriften van Frans Zwarts en Michael Moortgat
X-91-13 Max I. Kanovich — The Multiplicative Fragment of Linear Logic is NP-Complete
X-91-14 Max I. Kanovich — The Horn Fragment of Linear Logic is NP-Complete
X-91-15 V. Yu. Shavrukov — Subalgebras of Diagonalizable Algebras of Theories containing Arithmetic, revised version
X-91-16 V.G. Kanovei — Undecidable Hypotheses in Edward Nelson's Internal Set Theory
X-91-17 Michiel van Lambalgen — Independence, Randomness and the Axiom of Choice, Revised Version
X-91-18 Giovanna Cepparello — New Semantics for Predicate Modal Logic: an Analysis from a standard point of view

# The ITLI Prepublication Series

**1986** 86-01 The Institute of Language, Logic and Information
86-02 Peter van Emde Boas A Semantical Model for Integration and Modularization of Rules
86-03 Johan van Benthem Categorial Grammar and Lambda Calculus
86-04 Reinhard Muskens A Relational Formulation of the Theory of Types
86-05 Kenneth A. Bowen, Dick de Jongh Some Complete Logics for Branched Time, Part I Well-founded Time, Forward looking Operators
86-06 Johan van Benthem Logical Syntax
**1987** 87-01 Jeroen Groenendijk, Martin Stokhof Type shifting Rules and the Semantics of Interrogatives
87-02 Renate Bartsch Frame Representations and Discourse Representations
87-03 Jan Willem Klop, Roel de Vrijer Unique Normal Forms for Lambda Calculus with Surjective Pairing
87-04 Johan van Benthem Polyadic quantifiers
87-05 Víctor Sánchez Valencia Traditional Logicians and de Morgan's Example
87-06 Eleonore Oversteegen Temporal Adverbials in the Two Track Theory of Time
87-07 Johan van Benthem Categorial Grammar and Type Theory
87-08 Renate Bartsch The Construction of Properties under Perspectives
87-09 Herman Hendriks Type Change in Semantics: The Scope of Quantification and Coordination
**1988** LP-88-01 Michiel van Lambalgen *Logic, Semantics and Philosophy of Language:* Algorithmic Information Theory
LP-88-02 Yde Venema Expressiveness and Completeness of an Interval Tense Logic
LP-88-03 Year Report 1987
LP-88-04 Reinhard Muskens Going partial in Montague Grammar
LP-88-05 Johan van Benthem Logical Constants across Varying Types
LP-88-06 Johan van Benthem Semantic Parallels in Natural Language and Computation
LP-88-07 Renate Bartsch Tenses, Aspects, and their Scopes in Discourse
LP-88-08 Jeroen Groenendijk, Martin Stokhof Context and Information in Dynamic Semantics
LP-88-09 Theo M.V. Janssen A mathematical model for the CAT framework of Eurotra
LP-88-10 Anneke Kleppe A Blissymbolics Translation Program
ML-88-01 Jaap van Oosten *Mathematical Logic and Foundations:* Lifschitz' Realizabiility
ML-88-02 M.D.G. Swaen The Arithmetical Fragment of Martin Löf's Type Theories with weak $\Sigma$-elimination
ML-88-03 Dick de Jongh, Frank Veltman Provability Logics for Relative Interpretability
ML-88-04 A.S. Troelstra On the Early History of Intuitionistic Logic
ML-88-05 A.S. Troelstra Remarks on Intuitionism and the Philosophy of Mathematics
CT-88-01 Ming Li, Paul M.B.Vitanyi *Computation and Complexity Theory:* Two Decades of Applied Kolmogorov Complexity
CT-88-02 Michiel H.M. Smid General Lower Bounds for the Partitioning of Range Trees
CT-88-03 Michiel H.M. Smid, Mark H. Overmars, Leen Torenvliet, Peter van Emde Boas Maintaining Multiple Representations of Dynamic Data Structures
CT-88-04 Dick de Jongh, Lex Hendriks, Gerard R. Renardel de Lavalette Computations in Fragments of Intuitionistic Propositional Logic
CT-88-05 Peter van Emde Boas Machine Models and Simulations (revised version)
CT-88-06 Michiel H.M. Smid A Data Structure for the Union-find Problem having good Single-Operation Complexity
CT-88-07 Johan van Benthem Time, Logic and Computation
CT-88-08 Michiel H.M. Smid, Mark H. Overmars, Leen Torenvliet, Peter van Emde Boas Multiple Representations of Dynamic Data Structures
CT-88-09 Theo M.V. Janssen Towards a Universal Parsing Algorithm for Functional Grammar
CT-88-10 Edith Spaan, Leen Torenvliet, Peter van Emde Boas Nondeterminism, Fairness and a Fundamental Analogy
CT-88-11 Sieger van Denneheuvel, Peter van Emde Boas Towards implementing RL
X-88-01 Marc Jumelet *Other prepublications:* On Solovay's Completeness Theorem
**1989** LP-89-01 Johan van Benthem *Logic, Semantics and Philosophy of Language:* The Fine-Structure of Categorial Semantics
LP-89-02 Jeroen Groenendijk, Martin Stokhof Dynamic Predicate Logic, towards a compositional, non-representational semantics of discourse
LP-89-03 Yde Venema Two-dimensional Modal Logics for Relation Algebras and Temporal Logic of Intervals
LP-89-04 Johan van Benthem Language in Action
LP-89-05 Johan van Benthem Modal Logic as a Theory of Information
LP-89-06 Andreja Prijatelj Intensional Lambek Calculi: Theory and Application
LP-89-07 Heinrich Wansing The Adequacy Problem for Sequential Propositional Logic
LP-89-08 Víctor Sánchez Valencia Peirce's Propositional Logic: From Algebra to Graphs
LP-89-09 Zhisheng Huang Dependency of Belief in Distributed Systems
ML-89-01 Dick de Jongh, Albert Visser *Mathematical Logic and Foundations:* Explicit Fixed Points for Interpretability Logic
ML-89-02 Roel de Vrijer Extending the Lambda Calculus with Surjective Pairing is conservative
ML-89-03 Dick de Jongh, Franco Montagna Rosser Orderings and Free Variables
ML-89-04 Dick de Jongh, Marc Jumelet, Franco Montagna On the Proof of Solovay's Theorem
ML-89-05 Rineke Verbrugge $\Sigma$-completeness and Bounded Arithmetic
ML-89-06 Michiel van Lambalgen The Axiomatization of Randomness
ML-89-07 Dirk Roorda Elementary Inductive Definitions in HA: from Strictly Positive towards Monotone
ML-89-08 Dirk Roorda Investigations into Classical Linear Logic
ML-89-09 Alessandra Carbone Provable Fixed points in $I\Delta_0+\Omega_1$
CT-89-01 Michiel H.M. Smid *Computation and Complexity Theory:* Dynamic Deferred Data Structures
CT-89-02 Peter van Emde Boas Machine Models and Simulations
CT-89-03 Ming Li, Herman Neuféglise, Leen Torenvliet, Peter van Emde Boas On Space Efficient Simulations
CT-89-04 Harry Buhrman, Leen Torenvliet A Comparison of Reductions on Nondeterministic Space
CT-89-05 Pieter H. Hartel, Michiel H.M. Smid, Leen Torenvliet, Willem G. Vree A Parallel Functional Implementation of Range Queries
CT-89-06 H.W. Lenstra, Jr. Finding Isomorphisms between Finite Fields
CT-89-07 Ming Li, Paul M.B. Vitanyi A Theory of Learning Simple Concepts under Simple Distributions and Average Case Complexity for the Universal Distribution (Prel. Version)
CT-89-08 Harry Buhrman, Steven Homer, Leen Torenvliet Honest Reductions, Completeness and Nondeterminstic Complexity Classes
CT-89-09 Harry Buhrman, Edith Spaan, Leen Torenvliet On Adaptive Resource Bounded Computations
CT-89-10 Sieger van Denneheuvel The Rule Language RL/1
CT-89-11 Zhisheng Huang, Sieger van Denneheuvel, Peter van Emde Boas Towards Functional Classification of Recursive Query Processing
X-89-01 Marianne Kalsbeek *Other Prepublications:* An Orey Sentence for Predicative Arithmetic
X-89-02 G. Wagemakers New Foundations: a Survey of Quine's Set Theory
X-89-03 A.S. Troelstra Index of the Heyting Nachlass
X-89-04 Jeroen Groenendijk, Martin Stokhof Dynamic Montague Grammar, a first sketch
X-89-05 Maarten de Rijke The Modal Theory of Inequality
X-89-06 Peter van Emde Boas Een Relationele Semantiek voor Conceptueel Modelleren: Het RL-project
**1990** *Logic, Semantics and Philosophy of Language*
LP-90-01 Jaap van der Does A Generalized Quantifier Logic for Naked Infinitives
LP-90-02 Jeroen Groenendijk, Martin Stokhof Dynamic Montague Grammar
LP-90-03 Renate Bartsch Concept Formation and Concept Composition
LP-90-04 Aarne Ranta Intuitionistic Categorial Grammar
LP-90-05 Patrick Blackburn Nominal Tense Logic
LP-90-06 Gennaro Chierchia The Variablity of Impersonal Subjects
LP-90-07 Gennaro Chierchia Anaphora and Dynamic Logic
LP-90-08 Herman Hendriks Flexible Montague Grammar
LP-90-09 Paul Dekker The Scope of Negation in Discourse, towards a flexible dynamic Montague grammar
LP-90-10 Theo M.V. Janssen Models for Discourse Markers
LP-90-11 Johan van Benthem General Dynamics
LP-90-12 Serge Lapierre A Functional Partial Semantics for Intensional Logic