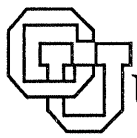
APPLICATION OF THE INTERACTIVE ACTIVATION MODEL TO DOCUMENT RETRIEVAL

Jonathan Bein & Paul Smolensky

CU-CS-405-88 May 1988



University of Colorado at Boulder
DEPARTMENT OF COMPUTER SCIENCE

Application of the Interactive Activation Model to Document Retrieval

Jonathan Bein & Paul Smolensky

CU-CS-405-88 May 1988

Department of Computer Science
Campus Box 430
University of Colorado
Boulder, Colorado 80309

Application of the Interactive Activation Model to Document Retrieval

Jonathan Bein & Paul Smolensky

CU-CS-405-88 May 1988

Department of Computer Science
Campus Box 430
University of Colorado
Boulder, Colorado 80309

Application of the Interactive Activation Model to Document Retrieval

Jonathan Bein
Paul Smolensky

University of Colorado at Boulder
Campus Box 430
Boulder, Colorado 80309

Submitted May 30, 1988

Abstract

In this paper we consider an application of the Interactive Activation Model [McClelland 82] to the problem of document retrieval. The issues in this application center around a neural net or "connectionist" model called Inductive Information Retrieval set forth in [Mozer 84]. The paper provides empirical results on the robustness of this model using a real-world document database consisting of 13,000 documents. In addition to results on robustness, we demonstrate the efficiency of this model both analytically and empirically.

Abstract

Dans cet article nous considerons une application du "Modele Interactif d'Activation" (Interactive Activation Model) [McClelland 82] au probleme de retrait de documents dans une base de donnees. Cette application est basee sur le reseau neuronal ou modele "connexioniste" dit "d'Acces Inductif a l'Information" (Inductive Information Retrieval) decrit dans [Mozer 84]. Dans l'article nous presentons des resultats empiriques sur la solidite de ce modele, obtenus en utilisant une base de donnees grandeur reelle contenant 13000 documents. En plus de ces resultats, nous demontrons l'efficacite du modele d'un point de vue analytique et d'un point de vue empirique.

1. Introduction

This paper describes research in progress regarding the application of neural network or "connectionist" methods to document retrieval. Typically, document retrieval is considered to be in the purview of standard information retrieval (IR) techniques. Work described in [Mozer 84, Belew 87] suggests that connectionist associative network can implement retrieval approaches that may be fruitfully applied to increase the quality in query results. The objective of the research reported in this paper is to rigorously test whether Mozer's Inductive Information Retrieval (IIR) model is useful with a real-world document database.

The objective in answering a query to a document database is to return *all* and *only* the relevant documents. If a retrieval system returns most of the relevant documents in the database, then it has high *recall*. If most of the documents returned by a retrieval system are deemed relevant, then it has high *precision*. In IR terminology, then, the goal is to maximize recall and precision. In practice, there is a tradeoff between precision and recall. The result is that users must work in an iterative fashion, submitting multiple queries, to obtain the desired documents.

In addition to the recall/precision tradeoff, there are three other important factors that complicate IR. First of all, queries as posed by users are often inconsistent and incomplete, resulting from a misunderstanding of the query system or more likely resulting from a lack of comprehension of the documents in the database. For example, a user may omit a keyword necessary to obtain the desired documents or might use the wrong keyword. Second of all, *indexing*, the method of establishing keywords for a given document, is prone to error whether automatic or manual methods are used. Hence, the document database is often inconsistent and incomplete too. Finally, the feedback and information resulting from one query may provide little help in determining how to iteratively refine the query.

From work in [Mozer 84] it appears that the connectionist approach has desirable properties with respect to these problems. The virtues of the IIR model are as follows:

- insensitivity to incorrect and incomplete queries - Here, keywords can be excluded or altered, yet the system still retrieves nearly the same set of documents.
- retrieval by example - Instead of (in addition to) specifying keywords, users may specify documents that are relevant. The system can determine from this example other similar documents to retrieve.
- induction of keywords - Given that queries are often incomplete and inconsistent, it is desirable for the system to be able to to determine other related keywords during execution.

These *induced* keywords can also be used to refine future queries.

- user profile - A connectionist IR system allows a system that is customized to the user's taste via learning algorithms. One form of customization involves setting the bias for certain documents to be higher than others.

However, a rigorous investigation of these properties has not yet been performed. The main issues which must be addressed are as follows:

- feasibility - Do the acclaimed properties of this model work with real document databases? In particular, does the model help to recover from incomplete and incorrect queries? Can the model actually induce additional documents and keywords?
- efficiency - Is there an implementation of this model which can run on conventional hardware that is both space and time efficient? If so, what are the limitations on the structure or size of the document base?
- utility - How useful is the additional functionality provided by this model to real users? How well does this model minimize the precision/recall tradeoff?

We consider the feasibility issue to be in the domain of connectionist research. The efficiency issue is an engineering problem. The utility of this model is primarily an IR problem. Work reported here will focus mostly on the feasibility and efficiency issues. The major contribution of this paper is two fold. First, we show that certain robustness properties of IIR hold in a real document database. Second, we provide strong evidence that production implementations of IIR are realistic from the standpoint of space and time efficiency. The next section describes the technical approach and architecture of the interactive activation model(IAM) for document retrieval. It covers related work from information retrieval as well as the IIR model and IAM. Section three discusses the efficiency of IIR. Section four covers a case study and other empirical testing of the model. The final section discusses conclusions and prospects for future research.

2. Technical Approach

At a high level, our approach to investigating feasibility and efficiency is to choose an existing document database, choose a connectionist simulator, and implement a model that performs associative retrieval. The advantage to using an existing document database is that we know that any results obtained are useful for at least one real world example. The tradeoff is that the results may not generalize to other document databases. The decision to use a simulator affects the total runtime, however, it will allow more rapid investigation of alternative architectures. It is our general belief that if by using a simulator the runtime is within an order of magnitude of the desired runtime that an efficient handcoded implementation will achieve the desired result. Also, part of the efficiency concerns will be addressed analytically, rather than empirically.

This project used the Symbolics Document Examiner(DEX), P3, and a connectionist IR model formulated by Mike Mozer. Each of these is described below.

2.1. Document Examiner

DEX (see [Walker 85]) is an on-line help system used on the Symbolics Lisp machines. It comprises all of the 12990 documents and about 6832 keywords in the 13 volumes of Symbolics system documentation. Thus, it includes information about Lisp functions, user commands, utilities, system implementation, hardware configuration, etc. Users submit queries to DEX (shown below in Figure 2-1) consisting of keywords by selecting the **Find** command. Names of documents that are relevant to that set of keywords are displayed in **Current Candidates** window. Users may view the contents of a candidate by selecting it. Results are displayed in the window on the left. Users may navigate through the document database by selecting highlighted hypertext in that window. This description of DEX merely illustrates how one commercial document database is seen from the user's perspective.

We have intentionally avoided any comparison with the documents retrieved by DEX and the documents retrieved in our model. From our standpoint, it is the size and structure of the document database that are relevant. Hence, an important aspect of DEX is that the database is not just a random collection of documents. Each document has a type associated with it. The types include flavor, function, variable, fragment, section, and presentation-type. In addition to the hypertext aspect described above, the database is hierarchically organized into chapters, sections, subsections, and so forth.

2.2. P3

P3 (see [Zipser 86]) is a general purpose connectionist network simulator developed at The University of California at San Diego. It contains four parts:

- The *plan language* allows the description of node types, node instances, and network topology.
- The *method language* provides a means for describing the behavior of a node through Lisp function calls.
- The *constructor* takes a plan and generates an executable connectionist model. Its role is similar to the role of a compiler in conventional programming languages.
- The *simulation environment* allows users to interactively execute a model to help refine the model or to actually run experiments. Hooks are provided to allow recording of vital data during simulation.

The simulation environment of P3 is shown in Figure 2-2, below. The squares correspond to

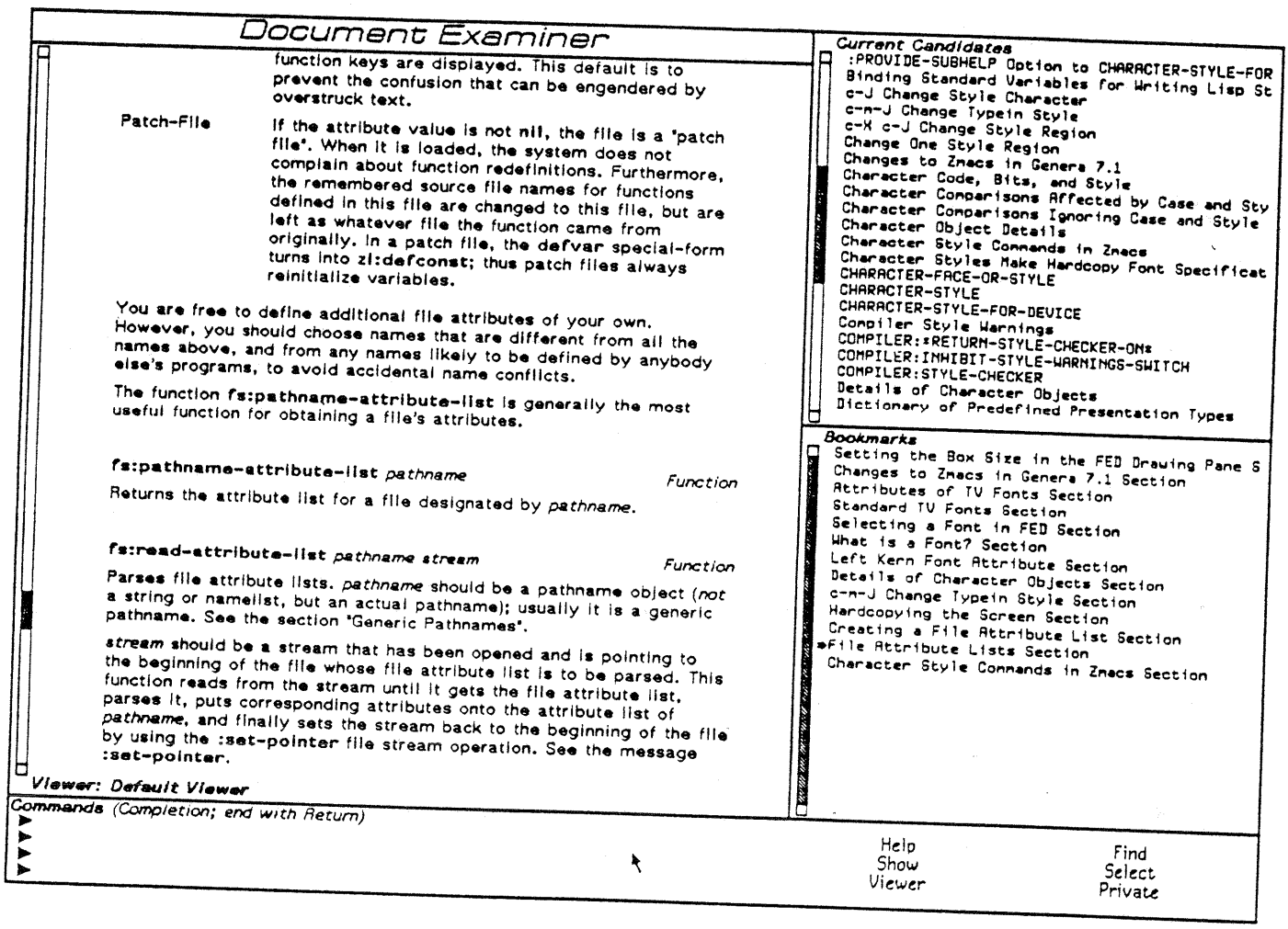


Figure 2-1: The Symbolics Document Examiner

documents and keywords in the model. Darker squares correspond to nodes with a higher activation level. Various attributes of any node may be changed interactively by selecting with the mouse. (In addition to the basic functionality provided by P3, we added functionality to P3 which allowed it to run as a subordinate process on the Lisp machine. This permitted batch simulations that derived our empirical results.) The implementation of IIR requires about six pages of P3 source code.

2.3. Information Retrieval

There are two basic models for IR: *boolean* and *vector*. In the boolean approach, queries are given to a system that include keywords and a list of boolean connectives. In the example below, the user is requesting any document that contains the keywords **computers** and **psychology** or the keyword **cybernetics**. These matching criteria are strict.

```
(computers and psychology) or cybernetics
```

In the vector model, each document has a *term weight* between 0 and 1 which tells how relevant that term is to the document. Queries are composed of *document term weights* which specify the ideal document from the user's standpoint. A function which measures the similarity between a document and the query is based on a normalized distance function: the smaller the distance, the better the match. An example of a vector query is given below.

```
computers: .56 psychology .39 cybernetics .95
```

2.4. Inductive Information Retrieval

In this subsection, we give a discussion of the connectionist IR approach. The description is at first conceptual, but is gradually refined. We will see that this model is a special case of IAM [McClelland82].

One disadvantage to the boolean model is that it doesn't use any partial matching criteria. Furthermore, there is no ranking of documents that are returned. The vector model ranks documents, but lacks some of the expressiveness of the boolean model. Neither model is good at handling inconsistent and incomplete queries or databases. A synthesis and general background on these models is given in [Salton 83a].

The connectionist inductive information retrieval(IIR) model is most similar to the vector model. It consists of two types of objects: documents and keywords. Between any document-keyword pair there is a connection with a weight of the keyword for that document. This is similar to the term weight described above. Also, between any document-keyword pair there is a symmetrical connection with a weight of the

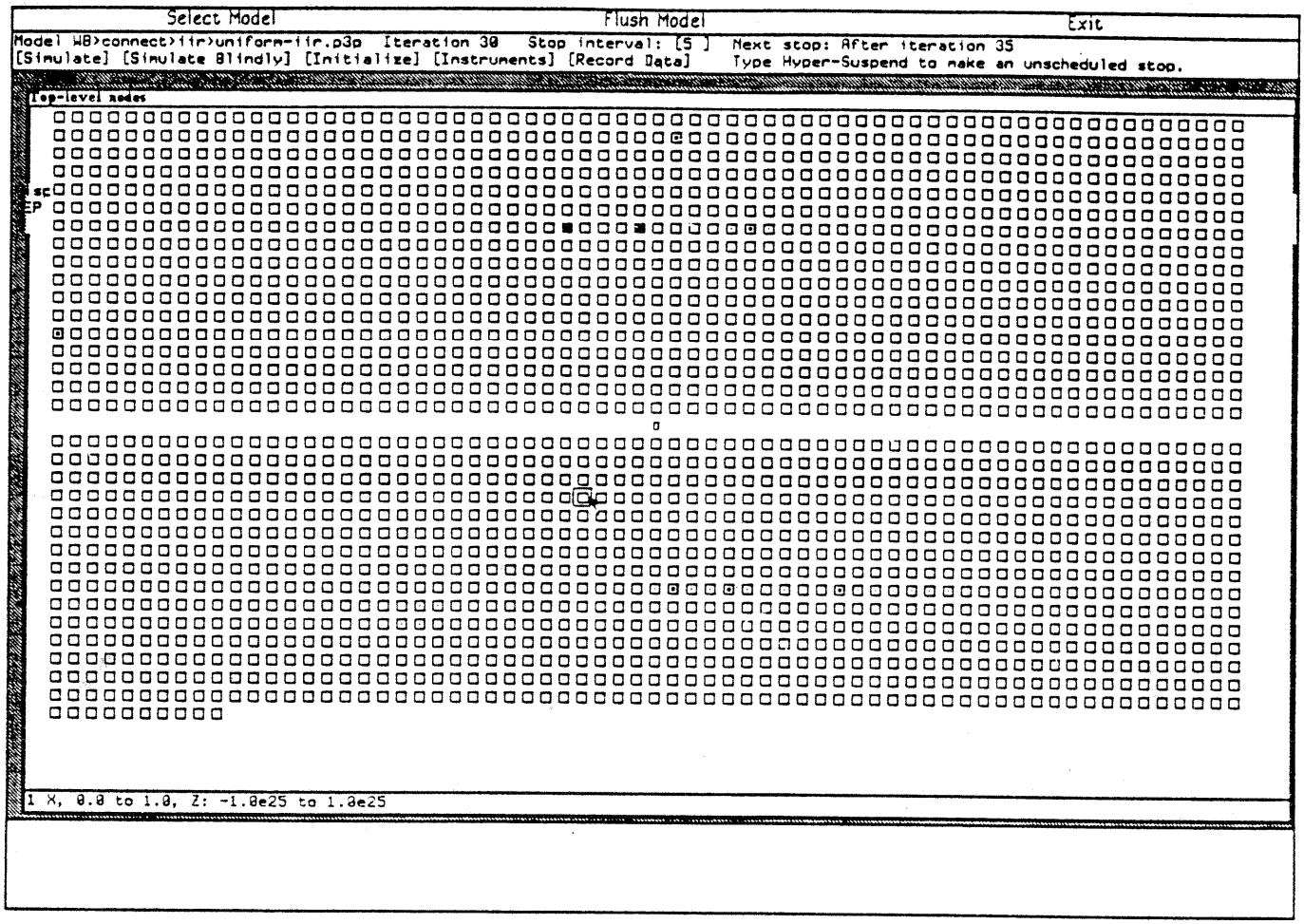


Figure 2-2: P3
Simulation Environment

document for that keyword. This has no analog in either the boolean or vector models. It is assumed that the number of connections is actually sparse, ie. there are few connections between documents and keywords.

Queries are expressed as in the vector model. By contrast, though, they are executed by passing activation back and forth from keywords to documents on each cycle. In the Figure 2-3 below, this corresponds to clamping the activity of the keywords (the bottom nodes.) In addition to the activation passed between the keywords and documents, there is a decay term associated with each document and each keyword. The net effect of this term is to drive the activation level back to the resting level (typically 0) whether the current level is positive or negative. After a certain amount of iteration, activation spreading stops, and documents that are more active are considered more relevant to the query. The crucial aspect of this approach is that other keywords may become activated in the iterative process of the query because the connections are bi-directional. Thus, keywords which were accidentally excluded may become activated via the transitive effects of activation spreading. This, in turn, activates other documents that would not have been reached in either the vector or boolean models. In this sense, the model is *inductive*. An asymmetric aspect of the model (see below) is that all documents contain inhibitory links to all other documents whereas keywords do not have mutually inhibitory connections. If one document is highly active, it will tend to inhibit other documents in the database. This competitive aspect helps keep the activation level of documents under control.

The IIR model is a special case of the IAM. It only has two levels instead of three or more. Also, it is not massively interconnected like IAM between any two levels. To date, we have made one extension to the IIR model, but have otherwise implemented it faithfully. The extension involves the weight on the inhibitory links between documents. As expressed in [Mozer 84] that weight is a constant. He notes that the value of this weight is one means for controlling the degree of associativity: the higher the weight, the less inductive retrieval. We have altered the model to gradually increase the level of inhibition over time. This has an effect similar to the cooling schedule in a simulated annealing process (see [Ackley 85, Kirkpatrick 83, Smolensky 86]) that allows a greater level of initial activity but ultimately settles.

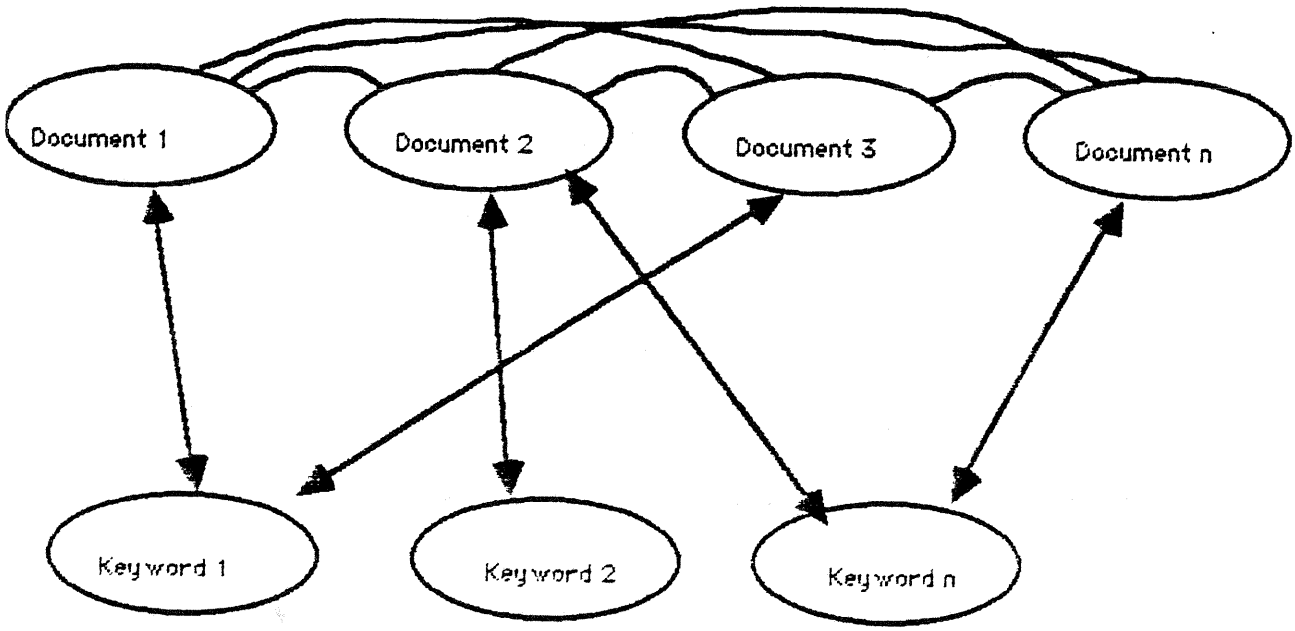


Figure 2-3: The Inductive Information Retrieval Model

3. Efficiency

In this section analytical and empirical results on the efficiency of IIR are provided.

The analysis of space efficiency in IIR is straightforward. The major issue is the number of connections, as the number of documents and keywords is of course linear. In the worst case, the number of connections between documents and keywords is equal to **2*ND*nd** where **ND** is the number of documents and **nd** is equal to the number of keywords. In this worst case, all documents have connections to all keywords. The number of inhibitory connections is simply **ND*ND**. Thus the total number of connections in the worst-case scenario is:

$$2*ND*nd + ND*ND = ND*(2*nd + ND)$$

For DEX, this would amount to 320,000,000 connections. This is an expensive implementation for a document database as small as DEX. Thus, larger size databases that include, e.g. only 100,000, documents are likely to be prohibitively expensive from both a time and space standpoint.

There are two encouraging facts regarding the efficiency of IIR. First of all, each document in a database does not have every possible keyword associated with it. In our experiments with DEX there were 6832 keywords, yet the largest number of keywords for a single document was 107. In fact, most documents had only four keywords associated with them. Out of a possible 156,000,000 excitatory connections in our experiments, there were only about 260,000. A second encouraging fact is that if a constant is used as the weight for lateral inhibition between documents, instead of a specific value on each connections, with a clever implementation, the number of connections diminishes to:

$$2*ND*nd + ND = ND(2*nd + 1)$$

The clever implementation involves using a single node to collect the total activity of all the documents. This value is then passed to each document to compute the lateral inhibition by subtracting out that document's activity. Thus, the space requirement to implement lateral inhibition is linear in the number of documents rather than quadratic. One question that must be addressed by future research is whether the heuristic just mentioned will impair the performance of the system by removing specific weights.

The time efficiency of IIR is a function of the number of connections also. Without providing a redundant analysis, we note that queries using 12990 documents required about five seconds per cycle. Our experiments used 60 cycles yielding a total query time of about five minutes. If we assume that a production implementation of IIR will not use P3 or any other simulator, but instead be manually coded to

exploit certain characteristics, then we conclude that IIR can be implemented efficiently.

4. Feasibility Results

In this section we look at empirical results. The method for our study is described first. Then we cover a case study that demonstrates the behavior of an individual query to IIR. Next, empirical results which demonstrate the auto-associative properties of IIR are presented. We have eschewed the study of precision/recall tradeoffs because they require a group of experts to agree on the relevant set of documents for a set of queries. Such studies have only been performed on a few document databases because this process is so labor intensive.

4.1. Method

All of the studies that were performed had the following common characteristics:

1. Each query was run for a total of 60 cycles.
2. Queries consisted of one to five keywords. Each query was chosen randomly from a corpus of actual queries that were submitted to DEX by users of all levels of sophistication.
3. A document was considered to have been selected if its final activity level was > .2. A keyword was considered to be induced if its final activity level was > .18. These values were derived informally from experience in a pilot study of IIR where the objective was to find a threshold that typically retrieved less than 100 documents
4. The document database consisted of all documents and keywords in DEX.
5. The inhibitory weight, **WDkDi**, was started at .007. A schedule which changed **WDkDi** by .0001 each cycle was also used.
6. Queries were expressed in P3 by clamping the value of query keywords to 1.0 via the extended P3 batch simulation routines.

4.2. A Case Study: "draw" and "circle"

This example is included because it demonstrates the retrieval properties IIR. The impressive part of this result is the rating of the documents and induced keywords. It correctly rates documents that are related to "draw" and "circle" higher than other documents. Interestingly, it rates documents related to cubic splines higher than documents related to straight lines.

Keywords	Activity
draw	1.0
circle	1.0
Graphics	.55662525
Option	.3299908
pattern	.30932835
PATTERN	.23046245
ANGLE	.23650017
RELAXATION	.21880357
cubic	.21880357
spline	.21880357
arrow	.2079927
DASHED	.19878739
:START	.19820416
:END	.18827803
:TOWARD	.18153937

Documents	Activity
:START-ANGLE TO GRAPHICS:DRAW-CIRCLE	.53690284
:END-ANGLE TO GRAPHICS:DRAW-CIRCLE	.53577447
:INNER-RADIUS TO GRAPHICS:DRAW-CIRCLE	.52248174
:START-RELAX TO GRAPHICS:DRAW-CUBIC-SPLINE	.5181
:END-RELAXATION TO GRAPHICS:DRAW-CUBIC-SPLINE	.4503905
:PATTERN-LEFT TO GRAPHICS:DRAW-PATTERN	.42877457
:PATTERN-TOP TO GRAPHICS:DRAW-PATTERN	.4181605
:PATTERN TO GRAPHICS:DRAW-RECTANGLE	.41572446
:HANDEDNESS TO GRAPHICS:DRAW-REGULAR-POLYGON	.40806454
:ARROW-HEAD-LENGTH TO GRAPHICS:DRAW-ARROW	.40101284
:RIGHT TO GRAPHICS:DRAW-PATTERN	.39721766
:ARROW-BASE-WIDTH TO GRAPHICS:DRAW-ARROW	.39117783
:BOTTOM TO GRAPHICS:DRAW-PATTERN	.38707647
:DASHED TO GRAPHICS:DRAW-ARROW	.38617364
:COPY-PATTERN TO GRAPHICS:DRAW-PATTERN	.3847595
:TOWARD-Y TO GRAPHICS:DRAW-STRING	.38426274
:TOWARD-X TO GRAPHICS:DRAW-STRING	.3690851
:DASHED TO GRAPHICS:DRAW-LINE	.3620239
:DASHED TO GRAPHICS:DRAW-LINES	.359678
:DRAW-END-POINT TO GRAPHICS:DRAW-LINE	.3583827
:STRETCH-P TO GRAPHICS:DRAW-STRING	.3524577

Table 4-1 - Query Results for of "draw" and "circle"

4.3. Empirical Results

As discussed in the introduction, the IIR model is supposed to be robust with the respect to the documents it retrieves given incorrect and inconsistent queries. A connectionist system that associates part of pattern with a larger completed pattern is an *auto-associator*. In IIR, one form of auto-association is inducing keywords that were not part of the user specified query. Another form of auto-association is the induction of documents that are not directly associated with the keywords from the user query. The empirical study is designed to test the auto-associative properties of IIR in two different scenarios. In the

first scenario, the study determines the sensitivity of IIR to *abridged queries* where a keyword is randomly deleted from the query. In the second scenario, the study determines the sensititity of IIR to *displaced queries* where a keyword is replaced by a morphological variant, e.g. replacing "graphics" by ":GRAPHICS". As mentioned previously, we do not have access to correct retrieval criteria for DEX. Instead, in this study, we define *relative precision* and *relative recall* as a function of the set of documents retrieved by IIR with the original unaltered query. In other words, the set of documents retrieved from the original query is considered to be completely correct.

For the abridged and displaced queries we measured relative precision and relative recall. The average values over 48 queries for these variables are summarized in the figures below.

	abridged	displaced
relative precision	.807	.664
relative recall	.839	.651

Figure 4-1: Average of Data for 36 Abridged and Displaced Queries Starting with Four Keywords

	abridged	displaced
relative precision	.796	.685
relative recall	.783	.736

Figure 4-2: Average of Data for 12 Abridged and Displaced Queries Starting with Three Keywords

Intuitively, it would seem that queries with a larger number of keywords are less sensitive to deletions than queries with a smaller number of keywords. In the extreme case, zero keywords, one retrieves zero documents. The data, displayed in the figures above, bear out this intuition: the relative precision and recall for the queries that initially had four keywords is higher than the relative precision and recall for queries that initially had three keywords.

For the displaced queries, the queries that had more keywords were more sensitive to displacement. An explanation for this result relies on the total activation in the system under different conditions. In general, the more keywords, the higher average level of activity of both documents and keywords. A network with a higher level of activity is more sensitive to the displacement of a single keyword than one that has a lower level of activity. This accounts for the greater sensitivity of the four keyword queries to displacement.

5. Conclusions and Future Directions

This section contains some very general conclusions about the model and a strategy to further investigate its feasibility, efficiency, and utility.

5.1. Conclusions

The research we have done validates the IIR model in general. It shows that the auto-associative characteristics described in [Mozer 84] scale up to a much larger database. Thus, IIR appears to be feasible for DEX. What is lacking are crisp results which constrain the conditions under which the feasibility of IIR increases or decreases. For example, it would be useful to know how well this approach works on databases with certain connectivity patterns that are not similar to DEX. The IIR approach also is efficient enough to permit use in production situations. We gave analytical and empirical evidence to support this claim. However, it is likely that some decisions which increase the feasibility of IIR may result in some performance penalties. Again, obtaining crisp results on feasibility will help to highlight such tradeoffs.

Below is a list of qualitative conclusions about the model which will affect future research in feasibility:

- Once a document or keyword has reached the artificial thresholds of acceptability that we imposed in our experiments, it tends to stay above that threshold. The activity may decrease, but usually not enough to exclude a document or keyword from inclusion in the final set. This suggests that lateral inhibition has a gradual effect on the model that does not result in wild oscillations.
- The rating assigned to a document does not have a consistent interpretation within queries and across queries. Moreover, there was a wide gap in the activity level of documents that were directly linked to keywords in the query versus those that weren't. If the user really wants the system to behave inductively, then the prescribed behavior should be that documents which are induced may have as high a rating as those that were not induced. On the other hand, the user might want the system to prefer documents which are not induced. This issue will be addressed in the utility studies. Finally, while it is true that the **rankings** produced by the IIR approach are pretty accurate, the **ratings** don't seem to be.
- [Mozer 84] describes an exponent that modifies excitatory input to documents and keywords. This exponent compensates for the bias of the system to prefer documents that have many keywords. However, if there is a wide variance in the number of keywords per document, it is hard to set the value of this exponent to appropriately address all situations. In our IIR version of DEX, certain documents with a large number of keywords often had a high activation independent of their relevance to the query. Controlling the built-in bias towards documents with a large number of keywords via such an exponent was our most serious difficulty with the model.
- The structure of DEX is extremely rich. Our experiments do not exploit all of the information in DEX pertaining to cross references and hierarchical structure. The current IIR model consisting of two levels needs to be expanded to look more like the IAM from which it was derived [McClelland 82]. In such a scheme, chapters, sections, subsections, documents, and keywords would each form a separate level. Activation for a document would tend to excite the subsection in which it was contained and so on all the way up to the chapter level.

Mutually excitatory links would be formed between documents that cross-reference one-another. In DEX there are currently some 5700 cross references.

- The convergence criteria in IIR need to be developed. As noted in section four, we used 60 cycles for all of our experiments. An equilibrium condition defined in terms of change in the activity of the documents and descriptors is the natural candidate.

5.2. Next Steps

There are two major issues that must be addressed. First, precise quantitative information about tradeoffs in the model need to be derived. The complications inherent in models with feedback suggest that an empirical approach is best. Specifically, we need to understand the effects of changing fan-in exponents, inhibition values, inhibition schedules, patterns of connectivity, and decay on relative precision and relative recall. Understanding the dynamics of these aspects will provide a firm basis for another sensitivity analysis: the importance of specific weights in the system. For example, using a specific weight between a keyword and a document is a form of knowledge. Yet, it may turn out that the value added by incorporating such a weight is minimal. If that is true, then the effort spent in deriving such a weight may be wasted. At this point, however, we cannot even say how one would establish weights although there are several candidate approaches that derive from the IR literature (see [Salton 83b].)

Once precise quantitative information is obtained, then experiments with actual users can be performed. These experiments should be designed to determine at least two things:

1. Does IIR provide a better retrieval mechanism in terms of recall/precision than the approach used in, e.g. DEX? Using a standard corpus is one way to analyze recall/precision. Real user feedback is potentially much better.
2. Can users make use of the output of IIR as a means for reformulating their query? It is claimed that the induced keywords will be useful to the user for such a purpose, but the claim is untested.

Acknowledgements

The author would like to gratefully acknowledge valuable conversations on information retrieval with Emilie Young and Steve Bulick of USWEST, Inc.. Mike Mozer helped clarify the few parts of his model that were not explicated in his paper. Hal Eden provided a tremenous computing environment. Without his help, this project could not have been accomplished. Thanks also to Isabelle Demuere and Olivier Brousse for the translation of the abstract.

This work has been supported in part by NSF grants IRI-8609599 and ECE-8617947 to the second

author, and by a grant to the second author from the Sloan Foundation's computational neuroscience program. This research was supported in part by Martin-Marietta under contract number 19X-CN981V (DOE contract number DE-AC05-840R21400).

Bibliography

[Ackley 85] - Ackley, D., Hinton, G., and Sejnowski, S., A learning algorithm for Boltzmann machines, Cognitive Science, 9, pp. 147-169, 1985.

[Belew 87] - Belew, R.L., Adaptive Information Retrieval: Machine Learning in Associative Networks, Doctoral Dissertation, University of Michigan, 1987.

[Kirkpatrick 83] - Kirkpatrick, S., Gelatt, C.D., and Vecchi, M., Optimization by simulated annealing, Science, 220, pp.671-680, 1983

[McClelland 82] McClelland, J.L. and Rumelhart, D.E., An Interactive Activation Model of Context Effects in Letter Perception: Part 1. An Account of Basic Findings, Psychological Review, Sept. 1981

[Mozer 84] - Mozer, M., Inductive Information Retrieval using Parallel Distributed Computation, ICS Report, UCSD, June 1984

[Salton 83a] - Salton, G., Fox, E.A., and Wu, H.. Extended Boolean Information Retrieval, CACM, 26, 1022-1036, 1983.

[Salton 83b] - Salton, G. and McGill, J. Introduction to Modern Information Retrieval, McGraw Hill, 1983.

[Hinton 86] - Hinton, G.E., and Sejnowski, T.J., Learning and Relearning in Boltzmann Machines, in Parallel Distributed Processing by Rumelhart and McClelland, Volume 1, 1986.

[Smolensky 86] - Smolensky, P., Foundations of Harmony Theory, in Parallel Distributed Processing by Rumelhart and McClelland, Volume 1, 1986.

[Walker 85] - Walker, J.H., Symbolics Document Examiner, SIGGRAPH Video Review, Vol. 19, 1985

[Zipser 86] - Zipser, D., Rabin, D. P3: A Parallel Network Simulating System, in Parallel Distributed

Processing by Rumelhart and McClelland, Volume 1, 1986.