Adaptive Information Retrieval:
Using a connectionist representation
to retrieve and learn about documents

Richard K. Belew

rik%cs@ucsd.edu

Cognitive Computer Science Research Group
Computer Science & Engr. Dept. (C-014)
Univ. California at San Diego
La Jolla, CA 92093

Abstract

AIR represents a connectionist approach to the task of information retrieval. The system uses relevance feedback from its users to change its representation of authors, index terms and documents so that, over time, AIR improves at its task. The result is a representation of the *consensual* meaning of keywords and documents shared by some group of users. The central focus goal of this paper is to use our experience with AIR to highlight those characteristics of connectionist representations that make them particularly appropriate for IR applications. We argue that this associative representation is a natural generalization of traditional IR techniques, and that connectionist learning techniques are effective in this setting.

1   Introduction

Artificial intelligence (AI) has provided a number of important knowledge representation ideas that have proven useful in information retrieval (IR) systems. Currently, connectionist networks (a.k.a. neural networks or parallel distributed processing (PDP) networks) are one of the most actively investigated representations in AI, and for this reason alone they may be of interest to IR researchers. However, this paper will argue that connectionist representations are particularly appropriate for the IR task. First, these networks naturally perform a type of *spreading activation* search that is

shown to be a natural extension of techniques typically used in IR systems. Second, powerful learning algorithms have been developed for connectionist systems that allow these representations to improve over time. This offers the potential of IR systems that automatically modify their indices to improve the probability of relevant retrievals.

We have investigated the use of connectionist techniques in an IR system called AIR. A query causes initial "activity" to be placed on some nodes, and this activity is allowed to propagate to other nodes until certain conditions are reached. Those nodes reaching the highest activity levels represent AIR's answer to the query. After the user has submitted a query and AIR has responded with a set of answers, the user labels some of the nodes as relevant and some as irrelevant. This "relevance feedback" causes new links to be created and weights on existing links to be changed. The feedback is averaged across many users to limit the impact of any one person's opinion. The result is a representation of the *consensual* meaning of keywords and documents shared by some group of users.

In the interest of brevity and because many excellent surveys already exist [12,23,13], this paper will assume a basic familiarity with connectionist representations and techniques. We begin with a description of the basic features of the AIR system's construction and operation. The central focus goal of this paper is to use our experience with AIR to highlight those characteristics of connectionist representations that make them particularly appropriate for IR applications. Section 3 relates AIR's associative retrieval behavior to more traditional approaches, and Section 4 presents evidence from experiments that suggest the system can also *learn* something worthwhile. We relate our work to several IR systems, and conclude with some of the future directions we see for research in using connectionist systems for IR.

2 The AIR system

The AIR[1] system represents a connectionist approach to the task of information retrieval. AIR uses feedback from its users (i.e., whether the documents retrieved were relevant or not) to change its representation of authors, index terms and documents so that, over time, AIR improves at its task. AIR's goal is to build a representation that will retrieve documents that are more likely to be relevant to queries. Our current implementation operates on a collection of approximately 1500 bibliographic citations to documents on the subject of artificial intelligence (AI). In addition to the critical indices assigned each document we also maintain information about each document's author(s). This has allowed us to experiment with how the clear, unambiguous *attributes* typical of databases can interact with the imprecise and vagarious keyword *descriptors* typical of IR systems. Other attributes (e.g., facts of publication) could also be maintained and would be treated analogously to authors.

This section will first discuss some details of the basic mapping from the IR problem into a connectionist representation, give an overview of AIR's retrieval process, and then sketch the system's learning algorithm. Elsewhere we have described the details of AIR's connectionist learning algorithms [2], the design of the system's interface [20] and the model of consensual semantics underlying our approach [3]; the most complete description of the system is still my thesis [1].

2.1 Mapping the IR domain into a connectionist representation

As with most connectionist systems, AIR uses a *weighted graph* as its basic representation. Unlike most connectionist systems, however, AIR does not begin from scratch (i.e., with a randomized network) but from a network constructed from an initial representation of the document, its authors and tentative keywords assigned to it. The goal of this initial representation is simply to make AIR's early retrievals reasonable enough that users will interact with the system. These interactions are then used to improve the initial representation. In fact, as we will argue in Section 3, even the initial, un-learned representation has desirable retrieval properties.

More specifically, each citation first causes a corresponding document node to be generated. An author node is then generated (if it doesn't already exist) for each author of the document. Our bibliographic collection has only the titles of each article, and virtually every word in the title is used as a tentative keyword.[2] A node is created for each of these keywords. Two links are then created between the document and each of its keywords (one in each direction), and two more between the document and each of its authors. Weights are assigned to these links according to an *inverse frequency* weighting scheme: the sum of the weights on all links going out of a node is forced to be a constant; in our system that constant is one.

The initial network is constructed from the superposition of many such documents' representations. Most of the experiments to be described in this report used a network constructed from 1600 documents, forming a network of approximately 5,000 nodes.

To the IR community, this automatic indexing procedure will seem simplistic. We use only titles (as opposed to larger samples of text such as abstracts or the full text of the document); our use of almost every word token in the documents' titles cannot scale to more realistic samples of text; and inverse frequency term weighting is arguably inferior to other methods [25]. Our goal is not to propose new automatic indexing techniques, however, but to build from the best of these that IR has to offer and extend them with our associative retrieval and adaptive techniques. AIR simply requires that the initial automatic indexing assign some weighted set of tentative keywords to each document. There are obviously many methods in IR for doing this; the procedure we used was merely the most straight-forward.

There is one property of the inverse weighting scheme on which AIR does depend, however. A network built using this keyword weighting scheme, together with similar constraints on the weights assigned author links, has the satisfying property of *conserving activity*. That is, if a unit of activity is put into a node and the total outgoing associativity from that node is one, the amount of activity in the system will neither increase nor diminish. This is very helpful in controlling the spreading activation dynamics of our network during querying. However, our understanding of the dynamical properties of connectionist systems has improved considerably since AIR's original design, and the conservative (in both senses) assumption motivating the inverse frequency weighting scheme can probably be relaxed to consider more reasonable IR methods.

---

[1] AIR stands for Adaptive Information Retrieval

[2] There are a few refinements, however. First, a "noise word" list is maintained; these are not indexed. Second, pluralized nouns are changed to their singular form. Punctuation and any numbers less than 100 are also removed, and all words are then capitalized.
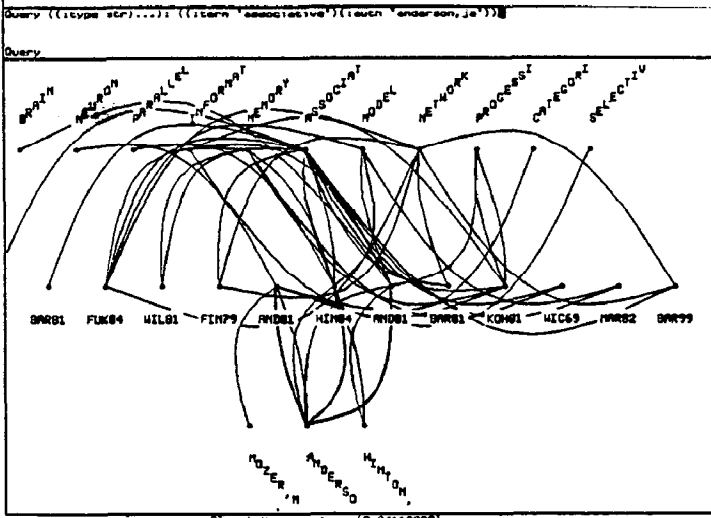
Figure 1: AIR interface

2.2 Querying and retrieval

Users begin a session with AIR by describing their information need, using a very simple query language. An initial query is composed of one or more clauses. Each clause can refer to one of the three types of "features" represented in AIR's network: keywords, documents or authors, and all but the first clause can be negated. This query causes "activity" to be placed on nodes in AIR's network corresponding to the features named in the query. This activity is allowed to propagate throughout the network and the system's response is the set of nodes that become most active during this propagation.

Figure 1 shows AIR's response to a typical query:

```
((:TERM ''ASSOCIATIVE'')(:AUTH ''ANDERSON,J.A.''))
```

This is the network of keywords, documents and author's considered relevant to this query. The nodes are drawn as a tri-partite graph, with keywords on the top level, documents in the middle and authors on the bottom. Associative links that helped to cause a node to become retrieved (and only those links) are also displayed. Heavier lines imply stronger weights. AIR uses directed links, and this directionality is represented by the concavity of the arcs; a clockwise convention is used. For example, a link from a document node (in the middle level) to a keyword node (in the top level) goes clockwise, around to the left.

Actually, this is only a picture of the *final state* of the system's retrieval. The network is actually drawn *incrementally*, with the first nodes to become significantly

active being drawn first and in the middle of the pane. As additional nodes become active at significant levels, they are drawn farther out along the three horizontal axes and the links through which they became active are drawn as well. We believe this dynamic display has at least two real advantages. First, the fact that AIR provides the first part of its retrieval almost immediately means that the user is not impatiently waiting for the retrieval to complete (typically 5-10 seconds in this implementation). Second, displaying the query's dynamics helps to give the user a tangible feeling of "direct manipulation" [20]; the user "prods" the network in a certain place, and then watches as waves of activity flow outward from that place.

Also, the regions immediately around each node are made "mouse sensitive" so that when the user puts the mouse near a node, more information about that node becomes visible in the "Who Line" at the bottom of the screen. The additional information is most useful for document nodes. These nodes are labeled with only a brief "citation" string: the first three letters of the first author's name are concatenated with the last two digits of the year of publication. The "Who line" shows both a complete citation (i.e., full names of all authors) and the title of the document. For keywords and authors, this information is simply the full keyword or author's name, which may have been truncated on the node's label.

2.3 Relevance Feedback

Queries subsequent to the first are performed much differently. After AIR is done retrieving the network of features, the user responds with *relevance feedback*, indicating which features are considered (by that user) relevant to the query and which are not. Using a mouse, the user marks features with the symbols: ++, +, − and − −, indicating that the feature was Very Relevant, Relevant, Irrelevant, or Very Irrelevant, respectively. Not all features need be commented upon.

The system constructs a new query directly from this feedback. First, terms from the previous query are retained. Positively marked features are added to this query, as are the negated versions of features marked negatively. Equal weight is placed on each of these features, except that features marked Very Relevant or Very Irrelevant are counted double.

From the perspective of retrieval, this relevance feedback becomes a form of *browsing*: positively marked features are directions which the user wants to pursue, and negatively marked features are directions which should be pruned from the search. From the perspective of learning, this relevance feedback is exactly the

training signal AIR needs to modify its representations through learning. This unification of learning (i.e., changing representations) and doing (i.e., browsing) was a central component of AIR's design. It mean that the collection of feedback is not an onerous, additional task for the user, but a natural part of the retrieval process.

2.4 Learning in AIR

Although AIR shares many features with other connectionist networks, it is sufficiently different from all of the "traditional" connectionist models (e.g., Hopfield nets [30], back-propagation nets [28]) that a unique learning algorithm was developed [2].

Nodes marked by the user with positive or negative feedback act as sources of a signal that then propagates backwards along the weighted links. A local learning rule then modifies the weight on links directly or indirectly involved in the query process. Several learning rules were investigated; the experiments reported here used a learning rule that correlated the activity of the "pre-synaptic" $node_i$ with the feedback signal experienced by the "post-synaptic" $node_j$:

$$w_{ij} \propto Corr(n_i\ active, n_j\ relevant)$$
$$= \frac{\mu_{a_i \cdot r_j} - \mu_{a_i} \cdot \mu_{r_j}}{\sigma_{a_i} \cdot \sigma_{r_j}}$$
$$= \frac{\Sigma(a_i \cdot r_j) - \frac{\Sigma a_i \Sigma r_j}{N}}{\sqrt{\Sigma a_i^2 - \frac{(\Sigma a_i)^2}{N}}\sqrt{\Sigma r_j^2 - \frac{(\Sigma r_j)^2}{N}}}$$

The correspondence between this probabilistic learning rule and the probabilisitic basis of IR can be pursued. First, we make a correspondence between the connectionist notion of *activity* and the IR notion of *relevance*:

> The activity level of nodes at the end of the propagation phase is considered to be a prediction of the probability that this node will be judged relevant to the query presented by the user.

This interpretation constrained the AIR system design in several ways (e.g., activity is a real number bounded between zero and one, query nodes are activated fully). AIR also allows negative activity, which is interpreted as the probability that a node is *not* relevant. The next step of the argument is to consider a link weight w_{AB} to be the *conditional probability* that $Node_B$ is relevant given that $Node_A$ is relevant. Next, this definition must be extended inductively to include indirect, transitive paths that AIR uses extensively for its retrievals.

$$W_{ij} = 1 / m$$
$$W_{ji} = \frac{(1 - {}^\bullet CONSERVE^\bullet)}{n_j + 1}$$
$$W'_{jk} = {}^\bullet CONSERVE^\bullet \bullet W_{jk}$$
$$m = \text{\# existing nodes}$$
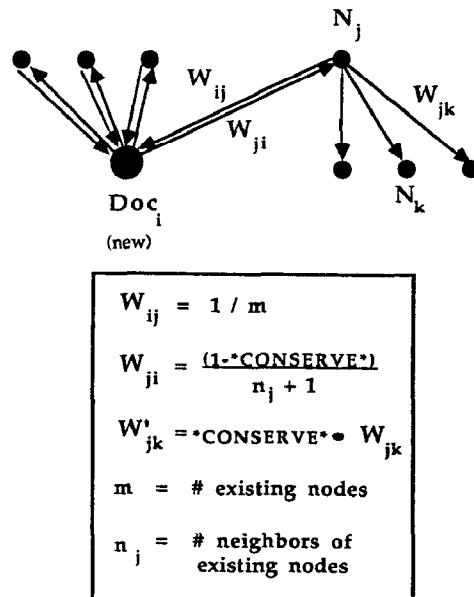$$n_j = \text{\# neighbors of existing nodes}$$

Figure 2: Incremental addition of new document

The system's interactions with users are then considered experiments. Given a query, AIR predicts which nodes will be considered relevant and the user confirms or disconfirms this prediction. These results update the system's weights (conditional probabilities) so as reflect the system's updated estimates. Thus, AIR's representation results from the combination of two completely different sources of evidence: the word frequency statistics underlying its initial indexing; and the opinions of its users.

Like most connectionist systems, the central focus of AIR's learning is the modification of weights on existing links. It is interesting to note, however, that the system can easily incorporate new documents and new query terms. First, a straight-forward mechanism exists for incrementally introducing new documents into AIR's database. Figure 2 shows the situation when a new document is to be added to an extent database. First, links are established from the new document to all of its initial keywords and to its authors; new keyword and author nodes are created as necessary. The weights on these links are distributed evenly so that they sum to a constant. So far, this procedure is identical with AIR's standard indexing scheme, used to incorporate the initial set of documents (see Section 2.1).

The only difference with incremental addition concerns the reciprocal links, from existing keyword and author nodes to the new document. Because the sum of the (outgoing) weights for all nodes is to remain constant, any associative weight to the new document must come from existing link weights. Figure 2 also shows the formula used to redistribute weights on existing links,

14

where a new parameter — *CONSERVATIVE* — has been introduced to control the weight given these new links at the expense of existing ones. If the network is untrained by users, this parameter can be set to zero so as to make the effect of an incremental addition exactly the same as if the new document had been part of the initial collection. In a trained network, setting *CONSERVATIVE* near unity insures that the system's experience incorporated in existing link weights is not sacrificed to make the new connections. Also, note that the computation required to place the new document is strictly *local*: only the links directly adjacent to the new documents immediate neighbors need be changed. The major observation about the inclusion of new documents, however, is that there is an immediate "place" for new documents in AIR's existing representation.

A second source of new information to the AIR system comes from users' queries. If a query contains a term unknown to AIR, this term is held in abeyance and AIR executes the query based on the remaining terms. Then, after the user has designated which of AIR's responses are relevant to this query, a new node corresponding to the new query term is created and becomes subjected to exactly the same learning rule used for all other nodes.

While easily incorporating new documents and new query terms are valuable properties for any IR system, from the perspective of machine learning these are both examples of simple *rote* learning, and necessarily dependant on the specifics of the IR task domain. The main focus of the AIR system is the use of general purpose connectionist learning techniques that, once the initial document network is constructed, are quite independent from the IR task.

3   Comparison with traditional IR techniques

While AIR is clearly performing an IR task, there are many ways in which it accomplishes this task in a distinctly non-standard way. The next three sections will argue that the system's representation, its selection of a retrieval set, and its interface are all natural generalizations of very standard IR techniques.

3.1   Generalized representation

Many parts of AIR's connectionist representation are not unfamiliar to IR researchers. Virtually all modern IR research (not to be confused with commercially
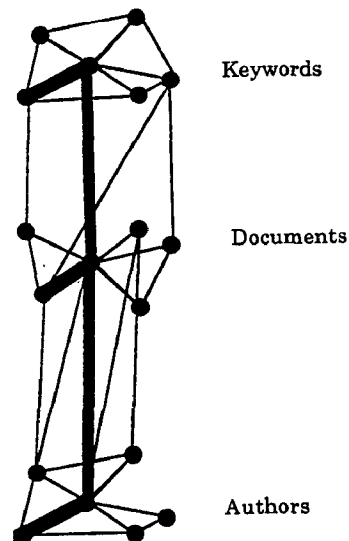

Figure 3: Higher-order associations used by AIR

vended IR systems) presupposes that the indexing relations between keywords and documents are weighted. It is also typical to consider similarity measures between pairs of nodes: clusterings between keywords or between documents, or automatic indexing between a keyword and a document.

Figure 3 portrays these pairwise associations within the broader context of the associative representation used by AIR. The system extends these pair-wise indexing and clustering relations to consider higher-order *transitive* relations as well. That is, if node A is associated with node B and node B is associated with node C, then node A is also considered to be associated with node C, but to a lesser extent.

Obviously, this transitive assumption is not always valid, and this may be why most IR research has not considered this extension. Recall, however, that AIR is an adaptive system. One of the critical problems facing any learning system is the generation of *plausible hypotheses*; i.e., theories which stand a better than average chance of being correct. Transitivity should therefore be considered a *default* assumption, the consequences of which will be subjected to selective pressures which favor appropriate transitivities and cull out inappropriate ones.

Note also that in our weighted, associative representation, the various semantics of the indexing and document- and keyword-clustering links are dropped in favor of a single, homogeneous "associative" relation. That is, AIR treats all three types of weighted links equally. If bibliographic citation data had been available, we would have used this valuable information as

15

well, but again lumped the semantics of these relations would have been dropped in favor of a simple associative weight. We contrast this *connectionist network* approach with others' use of spreading activation search in *semantic networks* in Section 5.2.

3.2 Generalized retrieval

Recall that AIR uses simple queries, specified by mentioning a set of features (keywords, authors, documents), or their negation. In particular, no provision is made for the traditional Boolean connectives AND and OR. Nevertheless, AIR's retrievel effectively emulates these logical operations. Because activity will spread from nodes mentioned by a query to their immediate neighbors first, the first nodes retrieved will be those that are in the *conjunction* of the query features. The next set retrieved are those in the *disjunction* of the query features. The point is that *the difference between AND and OR is a matter of degree*; this insight goes back to von Neumann.

The remaining nodes of the retrieval are not directly associated with either of the query features but are activated by other, indirect associations. It is this last class of *serendipitous* retrievals that are most impressive. While there is no obvious connection between the query and the retrieved document, a broad set of weak associations have cumulatively identified it as relevant. If the retrieval was appropriate, the user's positive feedback will make the connection between elements of the query and the document more direct; if the document was not relevant, this negative feedback will increase the associative distance (see Section 4).

AIR also provides a very convenient mechanism for varying from high-precision retrievals to high-recall retrievals. Varying a single parameter, viz., the threshold above which a node is considered to be "significantly active" enough to be retrieved, AIR can respond readily to varying user requirements for high precision or recall.

3.3 Generalized input and output

Also, the "input-output channel" from and to users has been widened by the AIR system. Typically, queries to IR systems are composed of keywords; it is also common to be able to specify authors of interest. But AIR also allows specification of *documents* in a query. The provision of this sort of this "query by example" seems a very useful extension language.[3]

---
[3]Mike Mozer was the first to point this out [15].

The result of AIR's retrieval is even more uncommon. The traditional result of an IR query is only documents (or more typically, citations to or proxies of documents). While this is AIR's major output as well, the system also provides keywords and authors. Keywords retrieved in this manner are considered "related terms" that users may use to pursue their searches. Retrieved authors are considered to be closely linked to the subject of interest.

It could be argued that these keywords and authors have no intrinsic value but are useful only to the extent that they ultimately lead to relevant documents. However, there are many ways in which a user might find related terms and centrally involved authors a valuable information product in their own right. For example, if a user wants to pursue their search in other information systems (such as a traditional library), these additional cues can be very useful. The fact that users had no more difficulty judging the relevance of keywords and authors than they did judging documents [1, Section 7.3.2] supports this view.

4 Adaptively warping associative distance

One way to describe the changes wrought by AIR's learning rule is in terms of how nodes in the network are effectively "moved" with respect to one another. For example, our initial indexing will cause keywords to be "close" to the documents they describe. We would like to have our learning algorithm move documents close to keywords that prove effective and away from poor ones, move related keywords closer together, move related documents closer together, etc.

More precisely, we can define a distance metric $ASSOC(x, y)$ over the nodes in our network. Two nodes x and x that are directly connected by a weighted link have $ASSOC(x, y) = w_{xy}$.[4] The associative distance between two nodes that are not directly connected is equal to the product of the weights of all links on the path connecting them, and summed over all such paths. Not only minimal paths are considered; the associativity due to shorter paths is merged with that due to longer ones. Potentially, this is an infinite computation (since AIR's net may contain cycles) so a maximum path length is imposed. The result is a measure of associativity (between two documents, for example) that captures a very large and rich but still tractible set of transitive associative relations. This is exactly the measure is imposed by AIR's retrieval process.

---
[4]Actually, this is only the first-order term of $ASSOC(x, y)$; longer paths must also be considered.

BEFORE   AFTER

#	Type	From			To			F->T	T->F	F->T	T->F
12	Phrase	3160	K	OBJECT	3528	K	ORIENTED	0.057	0.057	0.222	0.158
13	Re-writes	171	D	HOLLAND;76	158	D	HOLLAND;75	0.034	0.084	0.563	0.754
14	Re-writes	171	D	HOLLAND;76	170	D	HOLLAND;81	0.043	0.084	0.577	0.624
15	Re-writes	167	D	HOLLAND;80	159	D	HOLLAND;80	0.116	0.109	0.163	0.394
16	Re-writes	270	D	CARBONELL;83	269	D	CARBONELL;83	0.092	0.107	0.113	0.117
17	Re-writes	270	D	CARBONELL;83	1344	D	CARBONELL;83	0.095	0.097	0.114	0.12
18	Re-writes	961	D	FORBUS;83	995	D	FORBUS;84	0.189	0.152	0.186	0.352
19	Stem	2427	K	ADAPTATION	1904	K	ADAPTIVE	0	0	0.679	0.629
20	Stem	3447	K	LEARN	3346	K	LEARNING	0	0	0.074	0.1
21	Stem	2819	K	LAW	1491	K	LEGAL	0.009	0.016	0.322	-0.01
22	Stem	2819	K	LAW	1484	K	LAWYER	0	0	0.005	0
23	Stem	3308	K	MEDICAL	3495	K	MEDICINE	0	0	0.706	0.697
24	Synonym	3302	K	FRAME	3402	K	KNOWLEDGE	0	0.032	0	0.12
25	Synonym	2427	K	ADAPTATION	2695	K	EVOLUTION	0	0	0.903	0.639
26	Typo	4783	A	CARBONELL,JG	4784	A	CARBONELL,J	0.025	0	0.061	0
27	Typo	3768	A	SAMUEL,AL	3767	A	SAMUEL,AL	0.018	0.008	0.152	0.153
28	Typo	6	D	SAMUEL;63	339	D	SAMUEL,AL;99	0.085	0.085	0.185	0.204
29	Typo	6	D	SAMUEL;63	340	D	SAMUEL;67	0.111	0.17	0.2	0.178

Figure 4: Associative distance between nodes

While associativity is seems to be a sort of distance metric, there are two ways in which this interpretation is inappropriate. First, the $ASSOC(x, y)$ measure is a *directed* relation. That is, the associativity from x to y need not be the same as the associativity from y to x. Typical notions of distance do not have this asymmetry. Also, nodes can be *negatively* associated. The correlation learning rule used in AIR often cause negative, "inhibitory" connections between two nodes in a network. Thus, while the intuitively appealing notion of distance can help us understand what exactly is changing when AIR learns, we must be careful not to take that interpretation too seriously.

Some of the evidence of beneficial changes in AIR's representation as a result of its learning are summarized in Figure 4 in which the associative distance between pairs of nodes is given, both before the net had been trained and after.[5] Four particular classes of node pairs were evaluated. The first four pairs listed are examples of words which share a common "stem." IR has sophisticated algorithms for finding such stems, but any such algorithm is bound to be less than perfect. Even the untrained AIR system provided some of the benefits of stemming-like retrieval, simply as a consequence of the associative representation. What Figure 4 shows is that AIR's adaptive procedures moved to solidify these lexical relations. Words like "adaptive" and "adaptation," which initially had no association between them, now are connected by very strong associative links in each direction, simply because some users consistently found them co-relevant.

---

[5] AIR was trained by having 17 human subjects use the system for retrievals. The details of this experiment are contained in my thesis [1]

Similarly, example 5 shows how the same adaptive mechanisms have strengthened a phrase construct. The sixth pair is an example showing how these adaptive mechanisms can serve to correct errors. In this case, (due to a typographic error) two separate nodes were created to represent Arthur Samuel as an author. Users were able to recognize the misspelled version and evaluate it as co-relevant with the correct version. AIR's adaptive mechanisms then effectively "merged" these two nodes.

In truth, not all of the changes made by the system were this favorable. The primary reason for the counter-productive changes made by the system can be attributed to the small user population of our experiments. Because we only had access to 17 subjects, the sensitivity of our learning rule was set very high so that the changes made would be significant enough to perceive. Making dramatic changes in response to a single user's opinion, however, often resulted in modifications that proved inappropriate. AIR's learning rule is statistical and is therefore subject to the Law of Large Numbers. The system is designed make only very small changes in response to any one user, and depend on the statistical stability of large populations of such users to effect significant modification.

5   Related work

AIR was developed as a connectionist network, with the application of this technology to the IR domain coming as a fortuitous after-thought. In this respect it is similar to a system developed by Mike Mozer [15]. Mozer's project was a direct application of the parallel distributed processing (PDP) representation used by McClelland and Rumelhart for word perception [14,22].

17

Mozer used two levels of nodes, one corresponding to documents and one to keywords. The topology of connections between these nodes was quite different from AIR's, however. In particular, Mozer used mutually inhibitory links between every pair of documents. This was to form the sort of "winner take all" network used to pick a single alternative in the PDP model. In this application, however, this topology resulted in the retrieval of exactly one document, something we saw as a disadvantage. A second major difference is that Mozer's system did not learn.

As discussed in Section 3.1, however, AIR's representation has turned out to be closely related to other work in IR. Several IR researchers discussed the potential value of using general associative connections very early [9,8,28]. Below we mention three more recent IR projects that share important similarities with our work.

5.1 Document vector modification

One of the many facets of Salton's SMART project was an investigation of changes to documents' descriptions based on relevance feedback. This work was motivated by some of the same problems AIR attempts to address, viz., the augmentation of frequency-based indexing techniques with user evaluations, and an automatic procedure for allowing indices to track the inevitable changes that occur in natural language.

Friedman et al. describe a system in which the weights of keywords which discriminate well between relevant and non-relevant documents are increased on relevant documents and zeroed on non-relevant ones [11]. Brauen reports on a series of experiments in which the vector representation of all relevant documents $\vec{D}$ and some non-relevant documents $\vec{d}$, with respect to some query $\vec{q}$, are modified to move the relevant document's closer to $\vec{q}$ and non-relevant ones farther away:

$$\vec{D'} = \vec{D} + \alpha(\vec{q} - \vec{D})$$
$$\vec{d'} = \vec{d} + \alpha(\vec{d} - \vec{q})$$

where α is a constant set empirically to $\alpha = 0.2$ [5]. Several variations on this basic theme were investigated, and at least some of these proved quite successful.

Obviously, this work shares a great many similarities with AIR. Two features make it difficult to precisely characterize the differences with our work, however. First the SMART project, like virtually all IR research, uses what we view as an *omniscient* notion of relevance [3]. That is, in the omniscient view the relevance of a document with respect to any query is considered to

be absolute, determined by some "omniscient" observer. Both of the document vector modification algorithms mentioned above used queries for which the set of relevant documents had been determined *a priori*. AIR, on the other hand, uses the opinions of users that each had their own ideas about what was relevant. A key motivation for the use of connectionist learning algorithms in this context is their ability to find consistency in this very noisy relevance signal. Second, as described in Section 3.1, Salton's cosine correlation measure considers only direct, keyword-to-document associations while AIR makes use of a much wider web of indirect associations as well. To a first approximation the changes made by AIR to the direct keyword-to-document associations is not unlike those mentioned above, but AIR makes many other modifications as well.

5.2 Semantic networks

Another set of projects share with AIR the use of some form of spreading activation search. Preece has shown how many IR models can be replicated using the spreading activation mechanism [17]. He did this by giving each node logical processing capabilities, much like Fahlman's NETL system [10]. That is, individual nodes could be on or off, or could act differently, depending on the type and phase of different queries. Cohen and Kjeldsen have used a semantic network representation that uses this kind of logical processing to control the spread of activity through the network [6].

Salton and Buckley have analyzed the spreading activation search used in some of these systems and concluded that it is inferior to more traditional retrieval methods [24]. We believe that this conclusion does not apply to AIR for at least two reasons. First, they are correct in pointing out that when

> ... the relationships between terms or documents are specified by *labeled* links between the nodes .... the effectiveness of the procedure is crucially dependent on the *availability* of a representative node association map (p. 4,5) [Emphasis added]

That is, if spreading activation is to depend on the labels on the links of a semantic net, these labels must come from somewhere. In the systems mentioned above, this information (effectively, a refined sort of thesaurus) is programmed by the researcher, but this solution will not work for collections that cannot benefit from such manual analysis.

However, AIR is *not* a semantic network but a connectionist one. The difference between these two representations is admittedly a subtle one, but it is also

critically important [18]. One clear difference is that semantic networks make logical, deterministic use of *labeled* links, while connectionist networks like AIR rely on *weighted* links for probabilisitic computations. One consequence of this difference is that AIR is able to learn appropriate weights for its network while the labels on semantic networks must be programmed. A system like CRUCS, which incorporates *both* a semantic network and a connectionist network within a single system demonstrates that these need not be mutually exclusive soultions [4]. This system uses Derthick's $\mu KLONE$ [7] as the basis for an IR system that can make the logical inferences supported by KLONE and the partial matching allowed by a connectionist system.

Second, Salton and Buckley's own experiments show that when they modify a naive form of spreading activation search to include some of the techniques they have developed for their "vector process" searches — specifically, the use of L_2 keyword normalization and document length normalization — the performance of the two methods is quite comparable. As mentioned in Section 2.1, we are eager incorporate techniques like these into AIR's initial indexing algorithm and believe this can be done quite naturally.

5.3   Connection machine

Stanfill and Kahle have reported on an exciting new approach to the IR problem that exploits the massively parallel architecture of the Connection Machine (CM) [27], although the practical advantage of this approach over conventional sequential algorithms has been questioned [26,29]. Superficially, this approach might also seem similar to our project: the *Connection* Machine certainly sounds like it should have something to do with *connection*-ism. In fact, except for a shared interest in massively parallel computation, there is less overlap than might be expected. The problem is that the CM's SIMD (single instruction, multiple data) architecture does not suit the MIMD (multiple instruction, multiple data) computations typically required by connectionist models.

Another apparent similarity between AIR and the CM application is that both systems make central use of relevance feedback. However, the CM application makes only *transient* use of this feedback, to conflate a user's initial query into a much larger and more refined description. AIR uses relevance feedback to make *permanent* changes to its representation. It does appear possible, however, to incorporate some features of AIR in the CM retrieval process. An implementation of AIR on the CM is also being investigated.

6   Conclusions

We have described a connectionist approach to the IR problem. The associative, probabilistic nature of this representation is very well suited to the matching of queries with relevant documents, and the fact that there are powerful learning algorithms for connectionist networks allows these representations to improve with use. There are many variations on the connectionist theme; AIR is only one. Our experience suggests that other connectionist approaches to the IR problem are promising. Conversely, we can recommend the IR problem as a rich domain for connectionist researchers.

Our own work is attempting to extend the basic AIR system in several directions. In order to support the logical inferences typical of a legal IR application, we are merging AIR's connectionist network with a semantic network [19]. In an effort to make the probabilistic analysis of our system more sound, we are comparing AIR with a similar system based on Pearl's Bayesian Nets [16]. More philosophically, we are interested in developing a Wittgensteinian notion of natural language semantics that grows directly out of our representation of keywords [3], and also in comparing and contrasting the networks built by different social groups of users.

Acknowledgements

References

[1] R.K. Belew. *Adaptive information retrieval: machine learning in associative networks.* PhD thesis, Univ. Michigan, CS Department, Ann Arbor, MI, 1986.

[2] R.K. Belew. Designing appropriate learning rules for connectionist sytems. In *Proc. IEEE Intl. Conf. on Neural Networks*, June 1987.

[3] R.K. Belew and D.E. Rose. Learning semantics from word use. (unpublished manuscript), 1989.

[4] R.J. Brachman and D.L. McGuinness. Knowledge representation, connectionism, and conceptual re-

trieval. In *Proc. 11th Intl. Conf. on Research and Development in Info. Retrieval*, 1988.

[5] T.L. Brauen. Document vector modification. In G. Salton, editor, *The SMART retrieval system*, chapter 24. Prentice-Hall, New Jersey, 1969.

[6] P.R. Cohen and R. Kjeldsen. Information retrieval by constrained spreading activation in semantic networks. *Info. Proc. & Mgmt.*, 23(4):255–268, 1987.

[7] M.A. Derthick. A connectionist architecture for representing and reasoning about structured knowledge. In *Proc. 9th Conf. of the Cognitive Science Society*, July 1987.

[8] L.B. Doyle. Semantic road maps for literature searchers. *Journal of the ACM*, 1960.

[9] L.B. Doyle. Indexing and abstracting by association. *American Documentation*, 1962.

[10] S.E. Fahlman. *NETL: a system for representing and using real-world knowledge*. MIT Press, Cambridge, MA, 1979.

[11] S.R. Friedman, J.A. Maceyak, and S.F. Weiss. A relevance feedback system based on document transformations. In G. Salton, editor, *The SMART retrieval system*, chapter 23. Prentice-Hall, New Jersey, 1967.

[12] G.E. Hinton and J.A. Anderson. *Parallel models of associative memory*. Lawrence Erlbaum Assoc., Hillsdale, NJ, 1984.

[13] Geoffrey E. Hinton. Connectionist learning procedures. *Artificial Intelligence*, 1987. (to appear).

[14] J.L. McClelland and D.E. Rumelhart. An interactive activation model of the effect of context on language learning (part 1). *Psychological Review*, 1981.

[15] M.C. Mozer. Inductive information retrieval using parallel distributed computation. Technical report, Inst. Cognitive Science, UCSD, La Jolla, CA, 1984.

[16] J. Pearl. *Probabilistic Reasoning in Intelligent Systems*. Morgan Kaufmann, San Mateo, CA, 1988.

[17] S.E. Preece. *A spreading activation network model for information retrieval*. PhD thesis, CS Dept., Univ. Illinois, Urbana, IL, 1981.

[18] D.E. Rose and R.K. Belew. A case for symbolic/sub-symbolic hybrids. In *Proc. Cognitive Science Society*, 1989. (submitted).

[19] D.E. Rose and R.K. Belew. Legal information retrieval: a hybrid approach. In *Second Intl. Conf. on AI and the Law*, 1989. (to appear).

[20] D.E. Rose and R.K. Belew. Toward a direct-manipulation interface for conceptual information retrieval. In *Proc. ASIS Mid-year meeting*, 1989. (to appear).

[21] D.E. Rumelhart, G.E. Hinton, and R.J. Williams. Learning internal representations by error propagation. In J.L. McClelland and D.E. Rumelhart, editors, *Parallel distributed processing: Explorations in the microstructure of cognition*, chapter 8. Bradford Books, Cambridge, MA, 1986.

[22] D.E. Rumelhart and J.L. McClelland. An interactive activation model of the effect of context in language learning (part 2). *Psychological Review*, 1981.

[23] D.E. Rumelhart, J.L. McClelland, and PDP Research Group. *Parallel Distributed Processing: Explorations in the Microstructure of Cognition. Volume I: Foundations*. Bradford Books, Cambridge, MA, 1986.

[24] G. Salton and C. Buckley. On the use of spreading activation methods in automatic information retrieval. Technical Report 88-907, Dept. Computer Science, Cornell Univ., Ithaca, NY, April 1988.

[25] G. Salton and M.J. McGill. *Introduction to information retrieval*. McGraw-Hill, Inc., New York, NY, 1983.

[26] Gerard Salton and Chris Buckley. Parallel text search methods. *Comm. of the ACM*, 31(2), February 1988.

[27] Craig Stanfill and Brewster Kahle. Parallel free-text search on the connection machine system. *Communications of the ACM*, 29(12), December 1986.

[28] H.F. Stiles. The association factor in information retrieval. *Journal of the ACM*, 1961.

[29] Harold S. Stone. Parallel querying of large databases: A case study. *Computer*, October 1987.

[30] David W. Tank and John J. Hopfield. Collective computation in neuronlike circuits. *Scientific American*, 6(257), December 1987.