# Estimating analogical similarity by vector dot-products of Holographic Reduced Representations.

Tony A. Plate
Department of Chemical Engineering
University of British Columbia
Vancouver, BC, Canada V6T 1Z4
tap@epi.terryfox.ubc.ca [1]

October 25, 1995

[1] This paper reports work performed in the Department of Computer Science at the University of Toronto

Abstract

Models of analog retrieval require a computationally cheap method of estimating similarity between a probe and the candidates in a large pool of memory items. The vector dot-product operation would be ideal for this purpose if it were possible to encode complex structures as vector representations in such a way that the superficial similarity of vector representations reflected underlying structural similarity. This paper describes how Holographic Reduced Representations (HRRs) [Plate 1991] provide such an encoding. HRRs are a method for encoding nested relational structures as fixed width distributed representations, i.e., vectors. The similarity of HRRs that can be computed quickly using dot-products. If two HRRs involve similar objects and relations, their dot-product will be high and will be sensitive to some aspects of structural similarity. However, if the two HRRs do not involve similar objects their dot-product will be lower and will not be as sensitive to structural similarities. This paper also describes an enrichment of HRRs called "contextualization". The dot-product of contextualized HRRs is sensitive to more aspects of structural similarity, and is also sensitive to structural similarity in the absence of shared object features.

# 1 Introduction

Gentner and Markman [1993] suggest that the ability to deal with analogy will be a "watershed or Waterloo" for connectionist models based on distributed representations. They identify "structural alignment", which is the matching of corresponding components of two structures in a manner which preserves the relationships, as the central aspect of analogy making, and note the apparent ease with which people can do this in a wide variety of tasks. The main impediment to building distributed connectionist models for tasks such analogy evaluation has been the lack of an adequate distributed representation for nested compositional structure. Gentner and Markman wonder whether the surface forms of distributed representations for structured objects would be similar to the degree that the underlying structures were similar, but express pessimism about the prospects for the development of such a representation.

In this paper I describe how Holographic Reduced Representations (HRRs) [Plate 1991; Plate 1995], which are a fixed-width distributed representation for nested compositional structures, provide such a representation. In particular, they can be used to obtain fast estimates of similarity between structures. A HRR is a high dimensional vector, and the vector dot-product of two HRRs is an efficiently computable estimate of the overall similarity between the two structures represented. This estimate is sensitive to surface similarity and some, but not all, aspects of structural similarity,[1] even though correspondences between structure elements are not explicitly computed. Comparisons of HRRs can be made more sensitive to structural similarity by including additional bindings in the HRRs, which are derived using a technique I call "contextualization".

Gentner and Markman [1993] and Gentner, Markman and Wisniewski [1993] remark that people appear to perform structural alignment in a wide variety of tasks, including perception, problem solving, and memory recall. One task they and others have investigated is analog recall. A subject reads a number of stories and later reads a probe story. The task is to recall stories that are similar to the probe story (and sometimes evaluate the degree of similarity and perform analogical reasoning). This task is often thought to involve two stages – first is recall from long-term memory of stories that are potentially relevant and analogical to the probe story, and second is evaluation and use of the analogy, which involves construction of mappings between the recalled stories and the probe story. Two computer models of this process, Thagard *et al*'s [1990] ARCS (Analog Retrieval by Constraint Satisfaction) and Gentner and Forbus's [1991] MAC/FAC (Many Are Called, Few Are Chosen), also have two stages. The first stage selects a few likely analogs from a large number of potential analogs. The second stage searches for an optimal (or at least good) mapping between each selected story and the probe story and outputs those with the best mappings. MAC/FAC uses Falkenhainer, Forbus and Gentner's [1989] SME (Structure Mapping Engine) as its second stage. The two-stage architecture is illustrated in Figure 1. The first stage selects potential analogies to the probe from the large pool of items in long-term memory, using fast but superficial comparisons. The bottleneck depicted here is typical but not necessary – the computation required by the first stage could happen in parallel throughout long-term memory (e.g., in a content-addressable memory). The second stage performs a more expensive structural comparison between the probe and each potential analogy passed by the first stage. Two stages are necessary because it is too computationally expensive to search for an optimal mapping between the probe and all stories in memory. An important requirement for a first stage is that its retrieval speed scale well with both the size and number of episodes in long-term memory. This requirement prevented the authors of MAC/FAC and ARCS from having the first stage consider structural features, because they knew of no efficient way to do this.

The evidence that people take structural correspondences into account when evaluating and using analogies is longstanding and solid (see Gentner, Rattermann and Forbus [1993] for a review). However, it is less certain whether structural similarity influences access to long-term memory (i.e.,

[1] "Surface features" of stories are the features of the entities and relations involved, and "structural features" are the relationships among the relations and entities.

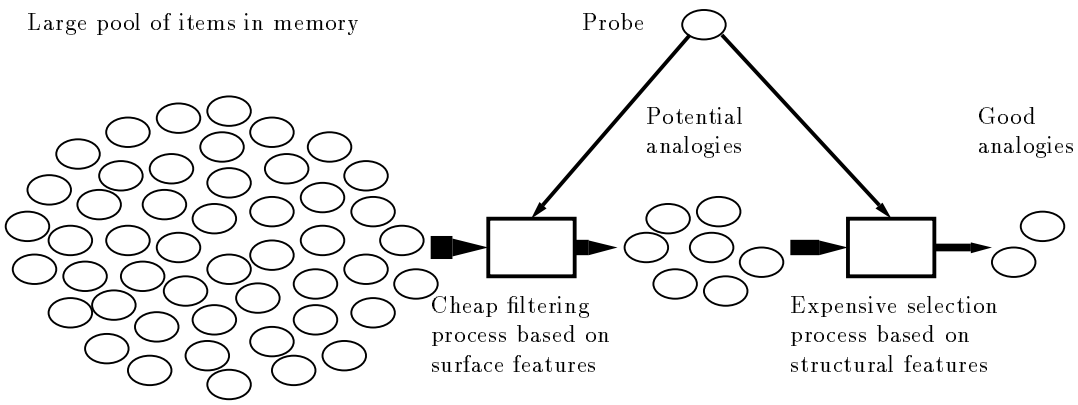



Figure 1: Two-stage models of analogy retrieval.

the first-stage reminding process). Gentner and Forbus [1991] and Gentner, Rattermann and Forbus [1993], found little or no effect of analogical similarity on reminding. Others, e.g., Wharton *et al* [to appear], Seifert and Hammond [1989] and Ross [1989], have found some effect. Wharton *et al* [to appear] suggest that the experimental design in earlier studies made it difficult to separate the effects of surface and structural similarity. In any case, surface features appear to influence the likelihood of a reminding far more than do structural features. Ross's [1989] study suggested that the effect of structural similarity on reminding is greater in the presence of surface similarity.

The first goal of this paper is to introduce HRRs as a distributed representations for nested compositional structure.

The second goal is to investigate how the dot-product of HRRs performs as an estimate of analogical similarity and to spell out the strengths and weaknesses of this estimate. To this end I show how dot-products of HRRs respond to various surface and structural aspects of similarity and compare the similarity judgements to those of ARCS and SME, two widely cited models of analogy evaluation and analogical reasoning. Thagard *et al*'s [1990] ARCS builds a local connectionist constraint satisfaction network (in its second stage) whose stable states are good interpretations of analogy between two structures. An interpretation of an analogy is a mapping of the entities in one structure to those in the other. The network is allowed to settle to a stable state and an overall similarity score is derived from how well the various constraints are satisfied. Falkenheimer, Forbus and Gentner's [1989] SME incorporates an explicit search for optimal analogical mappings. Both ARCS and SME provide an explicit interpretation of an analogy as well as a measure of goodness, and both can use the interpretation to perform analogical reasoning.

The third goal is to suggest that dot-products of HRRs could be an efficient mechanism for the first stage of a model of analog retrieval. This would be more a powerful mechanism than the current first stages of both ARCS and MAC/FAC because it would be sensitive to structural as well as surface similarity. However, a second stage would still be required for accurate evaluations of structural similarity, because the HRR dot-product is not an infallible measure of structural similarity. Furthermore, analogical reasoning tasks require an interpretation of an analogy, and this is not provided by HRR dot-products.

I use two sets of examples to illustrate how analogical similarity can be estimated by dot-products of HRRs. The first is a set of shape configurations from the paper by Markman *et al* [1993]. Figure 2 shows an example; the task is to judge which of A or B is more similar to X. These examples can each be represented by a single relation such as `above(triangle,circle)`. I include this set because it is a good illustration of some of the surface and simple structural features that people take into account when judging similarity. The similarity rankings given by HRR dot-products agree with

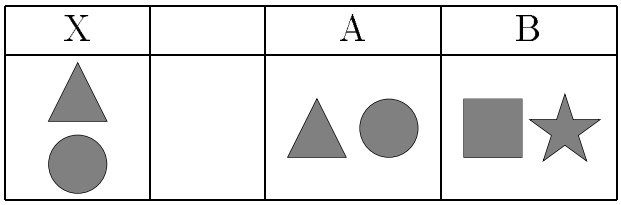



Figure 2: An example of the shape-configuration task.

those given by people.

The second set of examples consists of one-sentence episodes such as the following:

> "Spot bit Jane, causing Jane to flee from Spot."
> "Fido bit John, causing John to flee from Fido."
> "Rover bit Fred, causing Rover to flee from Fred."

The episodes in this set can be represented by two or three nested relations, e.g.:

```
cause(bite(spot,jane),flee(jane,spot)).
```

I use these examples to illustrate the extent to which HRRs are sensitive to the more complex aspects of structural similarity, such as patterns of variable instantiation and structural arrangement of propositions. It turns out that ordinary HRRs are only sensitive to structure in the presence of similarity among corresponding objects. However, HRRs can be made sensitive to structural similarity by incorporating contextual information.

In Section 2 of this paper, I describe HRRs, which are a distributed representation for nested compositional structure. In Section 3, I describe how HRRs can be applied to the shape-configuration examples. In Section 4, I describe how HRRs can be used to judge the similarity of episodes involving multiple, nested propositions. In Section 5, I describe how interpretations of an analogy, i.e., corresponding objects, can be extracted from HRRs. In Section 6, I discuss various issues, such as the relationship of HRR similarity ratings to ratings given by SME and ARCS, scaling to larger and more examples, flexibility of comparison, and pros and cons with respect to other distributed representations for nested compositional structure.

# 2 Holographic Reduced Representations (HRRs)

The difficulty of representing recursively nested structures has been an obstacle to the use of distributed representations and neural networks for higher-level cognitive tasks such a reasoning and language processing [Fodor and Pylyshyn 1988; Smolensky 1990; Hinton 1990]. There have been a number of methods proposed for forming associations in distributed representations in a manner able to handle recursively nested structure. These include Tensor Products [Smolensky 1990], RAAMs [Pollack 1990], and BoltzCONS [Touretzky 1990]. Holographic Reduced Representations (HRRs) are a distributed representation for nested compositional structures, which retain nearly all the advantages of distributed representations for objects. HRRs have much in common with Smolensky's tensor-products representation for propositions, but use circular convolution to associate roles and fillers, rather than the tensor product. The benefit of this is that the dimensionality of representations remains constant rather than increasing exponentially with the depth of composition as with

tensor products. Convolution has been used as the associative operator in a number of computational and psychological memory models [Willshaw, Buneman and Longuet-Higgins 1969; Borsellino and Poggio 1973; Liepa 1977; Willshaw 1981; Murdock 1982; Metcalfe Eich 1982; Murdock 1987; Paek and Psaltis 1987].

In this Section, I give an overview of HRRs; for more a detailed discussion, see Plate [1995] or Plate [1994].

## 2.1 Advantages of distributed representations

Distributed representations [Hinton, McClelland and Rumelhart 1986] are commonly described as patterns of activity over a large number of computing elements (neurons). For the purposes of this paper, I use an abstract version of this description: a distributed representation is a vector of real values. The number of elements in the vector (i.e., the vector dimension) is usually fixed.

Distributed representations are attractive for a number of reasons:

- There is a simple and computationally efficient measure of similarity between two distributed representations – the vector dot-product.[2] In general, similar items can be represented by vectors which have a high dot-product.
- Items are represented in a continuous space. This has two benefits:
  - It allows similarities to vary continuously.
  - Representations can be learned using gradient descent techniques.
- The information in the vector is distributed and redundant. This has several useful consequences:
  - Multiple items can be stored in a single buffer, by superimposing their representations. The number of items that can be superimposed is a soft-limit – items become degraded as more items are added.
  - The representations degrade gracefully as noise is added.
  - Representations which are degraded either by noise or by superposition can be cleaned up using an error-correcting associative memory.
- They can be processed in a parallel network of simple processing elements

## 2.2 Representational assumptions

The basic representational assumptions used in this work (and related work on distributed connectionist representations for nested structure) are as follows:

- All entities (including those with complex structure) are represented by fixed-width vectors.
- The representation of a compositional entity is derived, in a systematic fashion, from the representations of its components.
- The representation of an atomic entity (i.e., one not composed of other entities) is arbitrary. In HRRs, atomic or base-level entities are represented by random vectors.

[2] The vector dot-product of two $n$-dimensional vectors $\mathbf{x}$ and $\mathbf{y}$ is defined as: $\mathbf{x} \cdot \mathbf{y} = \sum_{i=0}^{n-1} x_i y_i$ where $x_i$ and $y_i$ are the $i$-th elements of $\mathbf{x}$ and $\mathbf{y}$. I assume that vectors are normalized, i.e., their Euclidean length is 1. If not, the cosine is often a better measure of similarity.

## 2.3 Associations and complex structure

For more complex tasks we require a way of representing associations between items. Suppose we have distributed representations for the concept "circle", "triangle", "small", and "large". We can represent a small circle by superposing the patterns for "small" and "circle". However, when we try to represent a small circle and large triangle we have the problem that the superposition of the four patterns is ambiguous – the information that small is associated with triangle and large associated with circle is lost – it could be a large circle and a small triangle. The same problem arises when we try to represent conceptual relations. Suppose we have a predicate representation for "Spot bit Jane": `bite(spot,jane)`. Spot is the agent of this relation, and Jane is the object (or patient). A distributed representation of this relation must be careful to preserve the information about which person is associated with which role (agent or object) so that there is no confusion with "Jane bit Spot". I refer to associations between roles and fillers as *role/filler bindings*.

One obvious way to avoid ambiguity about who is associated with which role is to divide the vector into several blocks, and devote a block to each role [Hinton 1990]. The drawback of this method is that it is unsuitable for representing recursively nested relations such as "Spot bit Jane, causing Jane to flee from Spot": `cause(bite(spot,jane),flee(jane,spot))`. An alternative is to use a function to combine two vectors (representing a role and a filler) into one. This is the approach taken by Smolensky [1990] with tensor products and Pollack [1990] with RAAMs and is also the approach taken with HRRs, though in each case the function used to combine vectors is different.

If the method for associating roles and fillers can be applied recursively, as it can with tensor products, RAAMs and HRRs, then it can represent nested structures. The main issue of concern in this paper, and the issue which concerned Gentner and Markman [1993] is whether the structure of complex objects is reflected in the surface form of the resulting representation.

## 2.4 Circular convolution

Circular convolution is an operation that maps two vectors onto one, i.e., $\circledast : \mathcal{R}^n \times \mathcal{R}^n \rightarrow \mathcal{R}^n$. If $\mathbf{x}$ and $\mathbf{y}$ are $n$-dimensional vectors (subscripted 0 to $n-1$), then the elements of $\mathbf{z} = \mathbf{x} \circledast \mathbf{y}$ are

$$z_i = \sum_{k=0}^{n-1} x_k y_{i-k}$$

where subscripts are taken modulo-$n$. Circular convolution can be viewed as a compression of the outer (or tensor) product of the two vectors, as Figure 3 shows. Each of the small circles represents an element of the outer product of $x$ and $y$, e.g., the middle bottom one is $x_2 y_1$. The elements of the circular convolution of $x$ and $y$ are the sums of the outer product elements along the wrapped diagonal lines.

Circular convolution can be regarded as a multiplication operator for vectors. It has many properties in common with both scalar and matrix multiplication. It is commutative ($\mathbf{x} \circledast \mathbf{y} = \mathbf{y} \circledast \mathbf{x}$), associative ($\mathbf{x} \circledast (\mathbf{y} \circledast \mathbf{z}) = (\mathbf{x} \circledast \mathbf{y}) \circledast \mathbf{z}$), and bilinear ($\mathbf{x} \circledast (\alpha\mathbf{y} + \beta\mathbf{z}) = \alpha\mathbf{x} \circledast \mathbf{y} + \beta\mathbf{x} \circledast \mathbf{z}$). There is an identity vector $\bar{I}$ ($\bar{I} \circledast \mathbf{x} = \mathbf{x}$) and a zero vector $\bar{0}$ ($\bar{0} \circledast \mathbf{x} = \bar{0}$). Inverses $\mathbf{x}^{-1}$ exist for most vectors ($\mathbf{x}^{-1} \circledast \mathbf{x} = \bar{I}$). Circular convolution can be computed in $O(n \log n)$ time using Fast Fourier Transforms.

An association between two items $\mathbf{x}$ and $\mathbf{y}$ is represented by the convolution of the two items: $\mathbf{x} \circledast \mathbf{y}$. The inverse vector of $\mathbf{x}$ can be used to reconstruct $\mathbf{y}$ from $\mathbf{x} \circledast \mathbf{y}$: $\mathbf{x}^{-1} \circledast \mathbf{x} \circledast \mathbf{y} = \mathbf{y}$. However, the inverse is numerically unstable (except under certain conditions) and not very useful in practice. For vectors which have randomly chosen elements independently distributed as $\mathrm{N}(0, 1/n)$ (the normal distribution with mean 0 and variance $1/n$) there is an approximate inverse with many attractive properties. The approximate inverse of $\mathbf{x}$ is denoted by $\mathbf{x}^*$, and is a simple rearrangement of

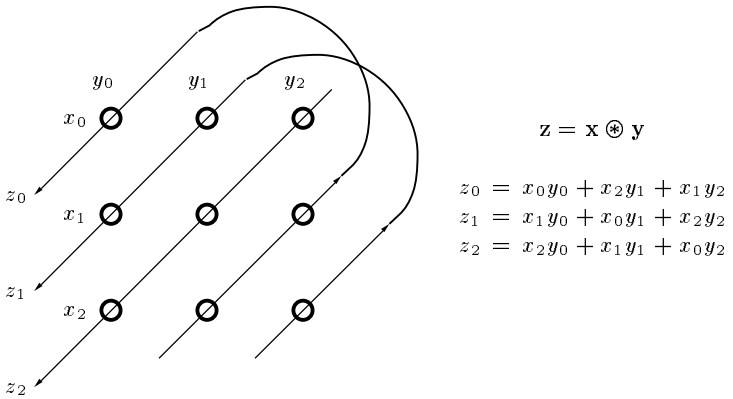



Figure 3: Circular convolution represented as a compressed outer product.

the elements of $\mathbf{x}$: $x_i^* = x_{-i}$, where subscripts are modulo-$n$. The approximate inverse is simple to compute and is numerically stable. The convolution product $\mathbf{x}^* \circledast \mathbf{x} \circledast \mathbf{y}$ can be written as $\mathbf{y} + \bar{\eta}$, where the $\eta$ can be considered as zero-mean noise whose magnitude (variance) decreases with increasing vector dimension.[3]

Multiple associations can be represented by the sum of the individual associations. For example, suppose $\mathbf{x}$, $\mathbf{y}$, $\mathbf{v}$, and $\mathbf{w}$ are all randomly chosen vectors with elements independently distributed as $N(0, 1/n)$. The association of $\mathbf{x}$ with $\mathbf{y}$ and $\mathbf{v}$ with $\mathbf{w}$ can be represented by $\mathbf{z} = \mathbf{x} \circledast \mathbf{y} + \mathbf{v} \circledast \mathbf{w}$. To find what is associated with $\mathbf{x}$ we convolve $\mathbf{z}$ with $\mathbf{x}^*$, the result is $\mathbf{x}^* \circledast \mathbf{x} \circledast \mathbf{y} + \mathbf{x}^* \circledast \mathbf{v} \circledast \mathbf{w}$. The first term here is approximately equal to $\mathbf{y}$ and the second term can be regarded as noise – it will not be highly correlated with any of $\mathbf{x}$, $\mathbf{y}$, $\mathbf{v}$, or $\mathbf{w}$. The sum of the two terms will be recognizable as a distorted version of $\mathbf{y}$. The vector dimension required is quite high – it must be in the hundreds for this to work well with a moderate number of associations.

If we want a convolution-based associative memory to provide accurate reconstructions of vectors, it must be equipped with an additional error-correcting auto-associative item memory. This is needed to clean up the noisy vectors retrieved from the convolution encodings. This item memory must store all the items that the system can produce. When given as input a noisy version of one of those items it must either output the closest item or indicate that the input is not close enough to any of the stored items. Note that we can store only a few associations as convolution encodings in a single vector, whereas we can store many vectors in the item memory. The item memory can be implemented in any number of ways, all that is required of it is that it perform a closest match retrieval of the vector presented to it. In the experiments described in this paper, the item memory is implemented as an exhaustive comparison over a list of vectors.

## 2.5 Constructing HRRs

An instantiated proposition can be represented by its label and a set of role/filler bindings. In HRRs, convolution is used to bind roles and fillers together. All the bindings and the predicate label are superimposed to give the representation of the entire proposition. The name of a predicate, and each role, are represented by separate vectors. In the following examples, three predicates are used: **bite**, with roles $\mathbf{bite}_{agt}$ and $\mathbf{bite}_{obj}$; **flee**, with roles $\mathbf{flee}_{agt}$ and $\mathbf{flee}_{from}$; and **cause**, with roles $\mathbf{cause}_{antc}$ and $\mathbf{cause}_{cnsq}$.

[3] $\mathbf{x}^* \circledast \mathbf{z}$ is known as the circular correlation of $\mathbf{x}$ with $\mathbf{z}$.

The HRRs for "Spot bit Jane." and "Jane fled from Spot." are

$$\begin{aligned} \mathbf{F}_{bite} &= \mathbf{bite} + \mathbf{bite}_{agt} \circledast \mathbf{spot} + \mathbf{bite}_{obj} \circledast \mathbf{jane} \qquad \text{and} \\ \mathbf{F}_{flee} &= \mathbf{flee} + \mathbf{flee}_{agt} \circledast \mathbf{jane} + \mathbf{flee}_{from} \circledast \mathbf{spot} \end{aligned}$$

where **bite** and **flee** are predicate labels, $\mathbf{bite}_{agt}$, $\mathbf{bite}_{obj}$, $\mathbf{flee}_{agt}$, and $\mathbf{flee}_{from}$ are roles, and **jane** and **spot** are fillers. All these components are $n$-dimensional vectors, as are $\mathbf{F}_{bite}$ and $\mathbf{F}_{flee}$, the representations for the entire propositions. This makes it possible to use a HRR as a role-filler in another proposition; e.g., in "Spot bit Jane, causing Jane to flee from Spot.":

$$\mathbf{F}_{cause} = \mathbf{cause} + \mathbf{cause}_{antc} \circledast \mathbf{F}_{bite} + \mathbf{cause}_{cnsq} \circledast \mathbf{F}_{flee}.$$

This HRR will be similar to other HRRs which have similar role/filler bindings or predicate labels. Often it is useful if the HRR is similar to other propositions which involve similar sub-propositions or entities in different roles. This can be done by incorporating the filler vectors, both object and propositional fillers, in the HRR, e.g.:

$$\mathbf{F}'_{cause} = \mathbf{cause} + \mathbf{jane} + \mathbf{spot} + \mathbf{F}_{bite} + \mathbf{F}_{flee} + \mathbf{cause}_{antc} \circledast \mathbf{F}_{bite} + \mathbf{cause}_{cnsq} \circledast \mathbf{F}_{flee}.$$

It is better not to do this in the HRRs for the lower level propositions, i.e., $\mathbf{F}_{bite}$ and $\mathbf{F}_{flee}$, because doing so would introduce undesirable bindings such as $\mathbf{cause}_{antc} \circledast \mathbf{spot}$ and $\mathbf{cause}_{antc} \circledast \mathbf{jane}$. HRRs for multiple propositions can be further enhanced by incorporating contextual information about fillers into bindings. I save the description of contextualization for Section 4.5.

## 2.6 Decoding HRRs

HRRs can be decoded using approximate inverses. The filler of a role in a HRR is extracted by convolving the HRR with the approximate inverse of the role. This gives a noisy version of the filler which can be cleaned up (if necessary) in an error-correcting associative memory. If we attempt to decode a non-existent binding in a HRR, the result is a noisy version of the zero vector. For example, with $\mathbf{F}_{bite}$ and $\mathbf{F}_{cause}$ as defined in Section 2.5, we get:

$$\begin{aligned} \mathbf{bite}^*_{agt} \circledast \mathbf{F}_{bite} &\approx \mathbf{spot} \\ \mathbf{bite}^*_{obj} \circledast \mathbf{F}_{bite} &\approx \mathbf{jane} \\ \mathbf{cause}^*_{antc} \circledast \mathbf{F}_{cause} &\approx \mathbf{F}_{bite} \\ \mathbf{bite}^*_{agt} \circledast \mathbf{cause}^*_{antc} \circledast \mathbf{F}_{cause} &\approx \mathbf{spot} \\ \mathbf{cause}^*_{cnsq} \circledast \mathbf{F}_{bite} &\approx \bar{0} \end{aligned}$$

The clean-up memory should contain all entities which are potential targets of decoding. This includes objects, such as **spot** and **jane**, roles, such as $\mathbf{bite}_{obj}$ and $\mathbf{cause}_{antc}$, predicate labels, such as **bite** and **cause**, and propositions of all levels, such as $\mathbf{F}_{bite}$ and $\mathbf{F}_{cause}$, but not individual role/filler bindings, such as $\mathbf{bite}_{agt} \circledast \mathbf{spot}$ or $\mathbf{cause}_{antc} \circledast \mathbf{F}_{bite}$.

## 2.7 Types and tokens

In many connectionist systems, the vectors representing items are analyzed in terms of "micro-features". Each element of the vector is considered to represent a simple feature. For example, Hinton's family trees network [Hinton 1990] learned micro-features representing concepts such as age and nationality. The requirement of HRRs that elements of vectors be randomly and independently distributed seems at odds with this interpretation. Furthermore, if every element of a vector is

regarded as a "micro-feature" it is unclear how to use the large number of them that the vectors of HRRs provide.

There is no requirement that single micro-features be represented by single bits in a distributed representation. Features also can be represented by high-dimensional distributed representations as wide as the representation of the whole object. An item having some features can be partly the sum of those features. Tokens of a type can be distinguished from each by the addition of some identity-giving vector that is unique for each token. Features can be represented by random vectors. For example, the person "John" can be represented by $\mathbf{john} = \mathbf{being} + \mathbf{person} + \mathbf{id}_{john}$, where $\mathbf{id}_{john}$ is some random vector that distinguishes **john** from representations of other people. Each component feature can be weighted according to its importance or salience, if necessary.

Advantages of this scheme over a local micro-feature representation are:

- The representation of any feature of an item will degrade gracefully as the elements of the vector representing the item are corrupted.
- The number of features in an item is only loosely related to the dimensionality of the vectors representing items.
- The vectors can be of as high a dimension as desired, and higher dimensionality will give better fidelity in the representation of features.
- The vectors representing items can be expressed as sums of vectors with random independently distributed elements.

When a set of vectors representing items is constructed from distributed features in this way the elements of the vectors will not be consistent with being drawn from independent distributions. However, if circular convolution is used to construct representations, all the expressions describing the recall and matching of vectors can be expanded to be in terms of the random feature vectors.

The idea of distributing features over the entire vector is not new. It is a linear transform and has been suggested by other authors under the name "randomization" or "random maps" [Schönemann 1987]. The equivalence of local and distributed patterns under linear operations was also noted by [Smolensky 1986].[4]

As mentioned before, care must be taken that the ownership of features is not confused when using this method to represent features (or attributes) of objects. It is interesting to note that if vectors consisting of superimposed features are convolved with distinct vectors (e.g., different role vectors) before being added, the results are not ambiguous:

$$(\mathbf{small} + \mathbf{triangle}) \circledast \mathbf{role}_1 + (\mathbf{large} + \mathbf{circle}) \circledast \mathbf{role}_2$$

is distinct from

$$(\mathbf{small} + \mathbf{circle}) \circledast \mathbf{role}_1 + (\mathbf{large} + \mathbf{triangle}) \circledast \mathbf{role}_2 .$$

Nonetheless, it may sometimes be necessary to convolve features with objects to avoid ambiguity, e.g., $\mathbf{small} + \mathbf{circle} + \mathbf{small} \circledast \mathbf{circle}$ for a small circle, and $\mathbf{person} + \mathbf{id}_{john} + \mathbf{person} \circledast \mathbf{id}_{john}$ for John. Further research is needed to determine if or when this is necessary.

[4] This does not mean that local and distributed patterns are equivalent under convolution – linear operations take only one argument – convolution takes two and is bilinear.

## 2.8 The expected similarity of convolution products

Convolution has the useful property that it is similarity preserving: two role/filler bindings tend to be similar to the extent that their components are similar. This works in a multiplicative fashion: the similarity of two bindings is the product of the similarities of the respective roles and fillers.[5] This property, together with the additive similarity-preserving property of superposition, is what causes HRRs to induce a natural similarity structure over propositions.

The bilinearity and randomizing properties of convolution make it possible to calculate analytically the expected similarity (dot-products) of convolution sums and products. The calculation is simple, if somewhat tedious.

For example, let $\mathbf{B}_1$ and $\mathbf{B}_2$ be two role/filler bindings involving similar entities:

$$\begin{aligned}
\mathbf{fido} &= \sqrt{0.5}\mathbf{dog} + \sqrt{0.5}\mathbf{id}_{fido} \\
\mathbf{rover} &= \sqrt{0.5}\mathbf{dog} + \sqrt{0.5}\mathbf{id}_{rover} \\
\mathbf{B}_1 &= \mathbf{bite}_{agt} \circledast \mathbf{fido} \\
\mathbf{B}_2 &= \mathbf{bite}_{agt} \circledast \mathbf{rover}
\end{aligned}$$

where $\mathbf{dog}$, $\mathbf{bite}_{agt}$, $\mathbf{id}_{fido}$, and $\mathbf{id}_{rover}$ are random *base* vectors with elements from $\mathrm{N}(0, 1/n)$. The scalars $\sqrt{0.5}$ are used to make the expected lengths of **fido** and **rover** equal to one. The vectors **fido** and **rover** are similar – their expected dot-product is 0.5.

We need several facts from Plate [1995] about the dot-products of vectors and convolutions of vectors with elements chosen randomly from $\mathrm{N}(0, 1/n)$:

$$\begin{aligned}
\mathrm{E}(\mathbf{a} \cdot \mathbf{a}) &= 1 \\
\mathrm{E}(\mathbf{a} \cdot \mathbf{b}) &= 0 \\
\mathrm{E}((\mathbf{a} \circledast \mathbf{b}) \cdot (\mathbf{a} \circledast \mathbf{b})) &= 1 \\
\mathrm{E}((\mathbf{a} \circledast \mathbf{b}) \cdot (\mathbf{a} \circledast \mathbf{c})) &= 0
\end{aligned}$$

The expected dot-product of $\mathbf{B}_1$ and $\mathbf{B}_2$ is

$$\begin{aligned}
&\mathrm{E}(\mathbf{bite}_{agt} \circledast \mathbf{fido}) \cdot (\mathbf{bite}_{agt} \circledast \mathbf{rover}) \\
&= \mathrm{E}((\mathbf{bite}_{agt} \circledast (\sqrt{0.5}\mathbf{dog} + \sqrt{0.5}\mathbf{id}_{fido})) \cdot (\mathbf{bite}_{agt} \circledast (\sqrt{0.5}\mathbf{dog} + \sqrt{0.5}\mathbf{id}_{rover}))) \\
&= \mathrm{E}(\sqrt{0.5}\mathbf{bite}_{agt} \circledast \mathbf{dog} \cdot \sqrt{0.5}\mathbf{bite}_{agt} \circledast \mathbf{dog} + \sqrt{0.5}\mathbf{bite}_{agt} \circledast \mathbf{dog} \cdot \sqrt{0.5}\mathbf{bite}_{agt}\mathbf{id}_{rover} \\
&\quad + \sqrt{0.5}\mathbf{bite}_{agt} \circledast \mathbf{id}_{fido} \cdot \sqrt{0.5}\mathbf{bite}_{agt} \circledast \mathbf{dog} + \sqrt{0.5}\mathbf{bite}_{agt} \circledast \mathbf{id}_{fido} \cdot \sqrt{0.5}\mathbf{bite}_{agt}\mathbf{id}_{rover}) \\
&= 0.5 + 0 + 0 + 0 = 0.5
\end{aligned}$$

In general, the expected similarity of two role/filler bindings is equal to the product of the expected similarities of their components.[6] The expected similarity of a binding to others with varying role and filler similarity is shown in Table 1. The vector **felix** is equal to $\sqrt{0.5}\mathbf{cat} + \sqrt{0.5}\mathbf{id}_{felix}$, and $\mathbf{cat}$, $\mathbf{id}_{felix}$, $\mathbf{flee}_{agt}$, and $\mathbf{chew}_{agt}$ are random base vectors.

So far I have only mentioned the expected dot-product of two vectors, whereas what is usually important is the actual dot-product. The dot-product is a fixed value for a particular pair of

[5] Provided that any role is not similar to any filler.

[6] Except when a particular base vector appears in both components of a binding, because $\mathrm{E}(\mathbf{a} \circledast \mathbf{a} \cdot \mathbf{a} \circledast \mathbf{a}) = 2 + 2/n$.

| Compared with $\mathbf{B}_1 = \mathbf{bite}_{agt} \circledast \mathbf{fido}$ | $E(\mathbf{bite}_{agt} \cdot \mathbf{bite}_{agt}) = 1.0$ | $E(\mathbf{bite}_{agt} \cdot \mathbf{chew}_{agt}) = 0.5$ | $E(\mathbf{bite}_{agt} \cdot \mathbf{flee}_{agt}) = 0.0$ |
|---|---|---|---|
| $E(\mathbf{fido} \cdot \mathbf{fido}) = 1.0$ | $E(\mathbf{B}_1 \cdot \mathbf{bite}_{agt} \circledast \mathbf{fido}) = 1.0$ | $E(\mathbf{B}_1 \cdot \mathbf{bite}_{agt} \circledast \mathbf{rover}) = 0.5$ | $E(\mathbf{B}_1 \cdot \mathbf{bite}_{agt} \circledast \mathbf{felix}) = 0.0$ |
| $E(\mathbf{fido} \cdot \mathbf{rover}) = 0.5$ | $E(\mathbf{B}_1 \cdot \mathbf{chew}_{agt} \circledast \mathbf{fido}) = 0.5$ | $E(\mathbf{B}_1 \cdot \mathbf{chew}_{agt} \circledast \mathbf{rover}) = 0.25$ | $E(\mathbf{B}_1 \cdot \mathbf{chew}_{agt} \circledast \mathbf{felix}) = 0.0$ |
| $E(\mathbf{fido} \cdot \mathbf{felix}) = 0.0$ | $E(\mathbf{B}_1 \cdot \mathbf{flee}_{agt} \circledast \mathbf{fido}) = 0.0$ | $E(\mathbf{B}_1 \cdot \mathbf{flee}_{agt} \circledast \mathbf{rover}) = 0.0$ | $E(\mathbf{B}_1 \cdot \mathbf{flee}_{agt} \circledast \mathbf{felix}) = 0.0$ |

Table 1:

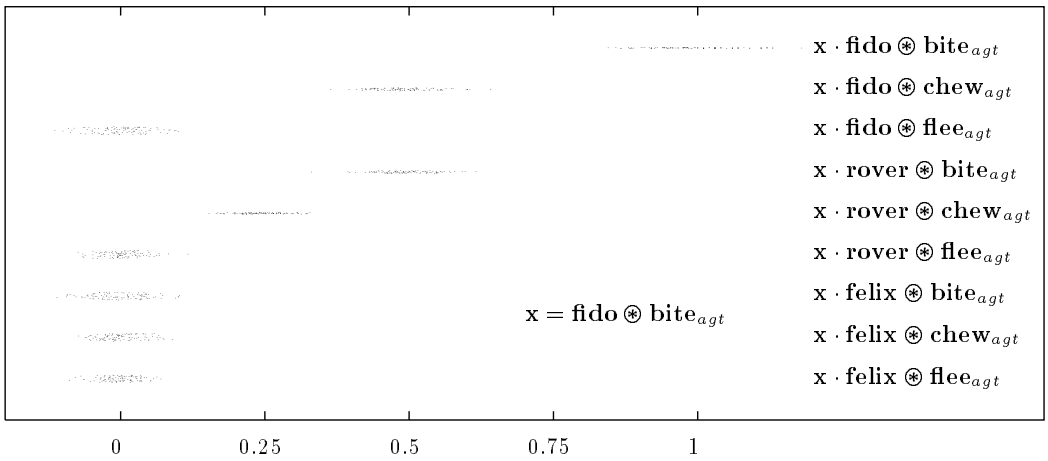



Figure 4: Observed dot-products of the role/filler binding $\mathbf{x} = \mathbf{fido} \circledast \mathbf{bite}_{agt}$ with various other bindings. Felix is a cat, and is not similar to Fido or Rover in the representation used.

vectors; the value depends on the values of particular elements of the vectors. However, we can usefully treat it as a random value that depends on the choice of random base vectors. The dot-products of superpositions and convolution products are normally distributed, and their variance is inversely proportional to $n$. The observed values of the dot-products in Table 1, for 100 choices of random base vectors of dimension 1024, are shown in Figure 4. The variance of each cluster is inversely proportional to the vector dimension, so for higher dimensional vectors, the clusters will be narrower.

## 2.9 Normalization

The magnitude (Euclidean length) of all vectors, (HRRs, tokens, and base vectors) should be around one. There are two reasons for this. The first is that we want the dot-product to reflect similarity rather than vector magnitude. The second is that when we superimpose two vectors, we want each to be equally represented in the result.[7]

There are two ways to accomplish this: weighting (as in the previous subsection) and normalization.[8] For all but simple examples it is easier to use normalization because of the difficulty of choosing the

[7] We can weight components if we want one to be more prominent in the sum.

[8] Normalization can be on the Euclidean length of the vector, as described here, or on the power of the frequency

appropriate weights. Furthermore, as I demonstrate in Plate [1994], normalization tends to reduce the variance of dot-products.

The normalized version of the vector $\mathbf{x}$ is denoted by $\langle \mathbf{x} \rangle$ and is defined as follows (the denominator is the Euclidean length of $\mathbf{x}$):

$$\langle \mathbf{x} \rangle = \frac{\mathbf{x}}{\sqrt{\sum_{i=0}^{n-1} x_i^2}}$$

### 2.10 Chunking

The idea of chunking, or breaking large structures in pieces of manageable size, fits together very well with HRRs. Chunking makes it possible to use HRRs to store structures of unlimited size, and HRRs provide a way of doing chunking that has some very attractive properties.

As more items and bindings are stored in a single HRR the quality of the items that can be extracted drops. If too many associations are stored the quality will be so low that the extracted items will be easily confused with similar items or, in extreme cases, completely unrecognizable. The number of items and bindings in a HRR grows with both the height and width of the structure being represented – $\mathbf{F}_{bite}$ has 3, while $\mathbf{F}_{cause}$ has 7 and $\mathbf{F}'_{cause}$ has 15. The number of items and bindings that can be stored for a given degree of quality of decoding grows linearly with the vector dimension [Plate 1995], but using vectors with very high dimension is not a satisfactory way to store large structures. A far superior way is to use chunk information. This involves storing sub-structures in the item memory, and using them when decoding components of complex structures. For example, to decode the agent of the cause antecedent of $\mathbf{F}_{cause}$ we first extract the cause antecedent. This gives a noisy version of $\mathbf{F}_{bite}$, so we then clean it up by accessing item memory and retrieving the closest match. Now we have an accurate version of $\mathbf{F}_{bite}$ from which we can extract the filler of the agent role. Using chunks during decoding like this reduces the effective number of items and bindings in HRRs and thus makes it possible to store structures of nearly unlimited size (though of course the number of chunks in items memory grows with the size of the structure).

Any system that uses chunks must have also have a way of referring, or pointing to the chunks. HRRs provide a very nice way of constructing chunks because a HRR is a pointer that tells us something about what it refers to. This is what the "reduced" in "Holographic Reduced Representation" refers to – a HRR is a compressed, or reduced version, of the structure it represents (the idea of a reduced description was introduced by [Hinton 1990]). It is somewhat misleading to call a HRR "a" pointer, because it "points" in multiple directions – to all its components (with suitable decoding). The advantage of having a pointer which encodes some information about what it refers to is that some operations can be performed without following the pointer. This can save much time. For example, we can decode nested fillers quickly if very noisy results are acceptable, or we can get an estimate of the similarity of two structures without decoding the structures. In the remainder of this paper I discuss the latter.

## 3 An experiment with shape configurations

Gentner, Markman, and Wisniewski [1993] report a simple experiment which demonstrates the importance of relational commonalities to human judgements of similarity. They asked subjects to judge the relative similarity of the pairs of configurations of geometric shapes shown in Table 2. Subjects were shown configuration X and then asked to say which of A or B was more similar to it. A and B were presented in a random order to each subject. The elements of each comparison are

components, as described in Plate [1994].

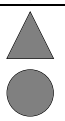
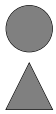
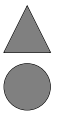
| | X | A | | B | |
|---|---|---|---|---|---|
| 1 | | | 10<br>(0.545)<br>[0.506 ± 0.032] | | 0<br>(0.347)<br>[0.335 ± 0.034] |
| 2a |  | | 8<br>(0.850)<br>[0.837 ± 0.017] | | 2<br>(0.529)<br>[0.506 ± 0.032] |
| 2b | | | 10<br>(0.656)<br>[0.666 ± 0.023] | | 0<br>(0.367)<br>[0.335 ± 0.036] |
| 3 | | | 10<br>(1.000)<br>[1.000 ± 0.000] |  | 0<br>(0.850)<br>[0.837 ± 0.017] |
| 4a |  | | 9<br>(0.817)<br>[0.831 ± 0.014] | | 1<br>(0.753)<br>[0.751 ± 0.020] |
| 4b | | | 9<br>(0.858)<br>[0.834 ± 0.017] | | 1<br>(0.745)<br>[0.753 ± 0.019] |
| 5 | | | 9<br>(1.000)<br>[1.000 ± 0.000] | | 1<br>(0.837)<br>[0.836 ± 0.014] |
| 6 | | | 10<br>(0.569)<br>[0.578 ± 0.028] | | 0<br>(0.369)<br>[0.415 ± 0.036] |

Table 2:

systematically varied in order to find out what aspects of the relations are important to similarity judgements. The item most frequently selected as more similar to X is shown in column A in all cases, and the number of times it was selected is shown as the top number beside (the numbers in the parenthesis and brackets are from experiments with HRRs, explained later).

Markman *et al* make the following comments about the stimuli:

> Item 1 suggests that having similar objects makes a pair of configurations similar, because only the similarity of the objects distinguishes between the two comparison figures. Items 2a and 2b suggest that having similar relations makes a pair similar regardless of whether the objects themselves are also similar. Item 3 suggests that having similar objects playing similar relational roles (e.g. both triangles on top) makes a pair similar. Items 4a and 4b suggest that any time a similar object plays a similar role it increases similarity, even if all of the objects taking part in the relation are not similar. Item 5 extends this finding by demonstrating that multiple relations can be used in the same comparison, as long as they give rise to the same object correspondences. In the most frequently selected alternative for this stimulus, the triangle is the top item and the smaller item, just as it is in the standard. Finally, Item 6 suggests that multiple relations can be matched, even when the objects making up the relations are dissimilar.

| | |
|---|---|
| | $\langle\langle \mathbf{lgCircle} + \mathbf{lgSquare}\rangle + \langle \mathbf{vertical} + \mathbf{above} \circledast \mathbf{lgCircle} + \mathbf{below} \circledast \mathbf{lgSquare}\rangle\rangle$ |
| | $\langle\langle \mathbf{lgCircle} + \mathbf{lgTriangle}\rangle + \langle \mathbf{horizontal} + \mathbf{left} \circledast \mathbf{lgCircle} + \mathbf{right} \circledast \mathbf{lgTriangle}\rangle\rangle$ |
| | $\langle\langle \mathbf{smStar} + \mathbf{lgSquare}\rangle + \langle \mathbf{vertical} + \mathbf{above} \circledast \mathbf{smStar} + \mathbf{below} \circledast \mathbf{lgSquare}\rangle\rangle$ |

Table 3:

> This simple study makes four central points. First, both relational similarities and object similarities contribute to similarity. Second, a relational match can be made whether or not the objects that make up the relation also match. Third, similarity is sensitive to the *bindings* between relations and objects that make up the relation. Finally, many different relations can contribute to similarity as long as these relational matches give rise to the same object correspondences.

I constructed simple Holographic Reduced Representations for each of the configurations, and then obtained estimates of similarity by computing the dot-product of the HRRs. Table 3 shows representative HRRs. Two predicates are used. One is "are horizontal", which has two roles: "left" and "right". The other predicate is "are vertical", which has two roles: "above" and "below". These predicate names and roles are represented by the vectors **horizontal**, **left**, **right**, **vertical**, **above**, and **below**. All of these are "random base vectors" (vectors with elements chosen independently from $\mathrm{N}(0, 1/n)$). The vectors representing the shapes are derived from the random base vectors **large**, **small**, **triangle**, **circle**, **square**, and **star** as follows ($\langle\cdot\rangle$ is the normalization operation):

$$\begin{array}{ll} \mathbf{smTriangle} = \langle \mathbf{small} + \mathbf{triangle}\rangle & \mathbf{lgTriangle} = \langle \mathbf{large} + \mathbf{triangle}\rangle \\ \mathbf{smCircle} = \langle \mathbf{small} + \mathbf{circle}\rangle & \mathbf{lgCircle} = \langle \mathbf{large} + \mathbf{circle}\rangle \\ \mathbf{smSquare} = \langle \mathbf{small} + \mathbf{square}\rangle & \mathbf{lgSquare} = \langle \mathbf{large} + \mathbf{square}\rangle \\ \mathbf{smStar} = \langle \mathbf{small} + \mathbf{star}\rangle & \mathbf{lgStar} = \langle \mathbf{large} + \mathbf{star}\rangle \end{array}$$

In these representations I treat size and shape as features and superimpose them to represent an object of a particular shape and size. This can create problems if the representations for objects are superimposed, because ownership of features becomes ambiguous, as discussed in Section 2.7. For example, the superposition $\mathbf{lgCircle} + \mathbf{smSquare}$ is the same as $\mathbf{smCircle} + \mathbf{lgSquare}$. However, this problem does not arise in these examples. The representation for "large circle above small square" is not the same as the representation for "small circle above large square", because the size features ("small" and "large") are bound to the roles "above" and "below".

I did not represent Items 5 and 6 with multiple propositions, contrary to Markman *et al*'s interpretation of their significance. In fact, if the smallness and largeness of the shapes in Items 5 and 6 were represented with a predicate "is larger than" rather than features, and simple HRRs constructed, the resulting dot-product ranking for Item 6 would not agree with the human judgements, unless contextualization was used. I discuss the relationship between features and predicates in more depth in Section 6.3.

In all cases the HRR similarity estimates give the same judgements of relative similarity as the human subjects. The HRR dot-products from one run (with 512-dimensional vectors) are shown in parentheses in Table 2 (e.g., 0.643 is the dot-product of the HRRs for A and X in the first row). The means and standard errors over 100 runs are shown in square brackets.

Probe: Spot bit Jane, causing Jane to flee from Spot.

| Episodes in long-term memory: | | Aspects of similarity: OA | FOR | HOR | RFB | HOS | OLI | Type |
|---|---|---|---|---|---|---|---|---|
| **E**1: | Fido bit John, causing John to flee from Fido. | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | LS |
| **E**2: | Fred bit Rover, causing Rover to flee from Fred. | ✓ | ✓ | ✓ | × | ✓ | ✓ | $AN^{cm}$ |
| **E**3: | Felix bit Mort, causing Mort to flee from Felix. | × | ✓ | ✓ | × | ✓ | ✓ | $AN_1$ |
| **E**4: | Mort bit Felix, causing Felix to flee from Mort. | × | ✓ | ✓ | × | ✓ | ✓ | $AN_2$ |
| **E**5: | Rover bit Fred, causing Rover to flee from Fred. | ✓ | ✓ | ✓ | $\frac{1}{2}$ | ✓ | × | $SS^{\times I}$ |
| **E**6: | John fled from Fido, causing Fido to bite John. | ✓ | ✓ | ✓ | ✓ | × | ✓ | $SS^{\times H}$ |
| **E**7: | Mort bit Felix, causing Mort to flee from Felix. | × | ✓ | ✓ | × | ✓ | × | $FA^{\times I}$ |
| **E**8: | Mort fled from Felix, causing Felix to bite Mort. | × | ✓ | ✓ | × | × | ✓ | $FA^{\times H}$ |
| **E**9: | Fido bit John, John fled from Fido. | ✓ | ✓ | × | ✓ | × | ✓ | $SS^{-H}$ |
| **E**10: | Fred stroked Rover, causing Rover to lick Fred. | ✓ | × | ✓ | × | × | × | $OO_1$ |
| **E**11: | Fred stroked Rover, Rover licked Fred. | ✓ | × | × | × | × | × | $OO_2$ |

Table 4:

# 4 Analogies between multiple relations

To investigate how well the dot-product of HRRs estimates the similarity of complex descriptions I used a set of stories each involving two or three nested relations. Because these stories are so simple, I prefer to refer to them as "episodes." The probe and the "memorized" episodes are shown in Table 4. I explain the acronyms for the aspects and types of similarity in the next two subsections. The memorized episodes are similar in different ways to the probe episode. These episodes are adapted from an example in Thagard *et al* [1990]. I added higher-order relational structure (the "cause" relation) and created new episodes.

In these episodes Jane, John, and Fred are people, Spot, Fido, and Rover are dogs, Felix is a cat, and Mort is a mouse. All of these are *objects* (also referred to as *entities*). Tokens of the same type are considered to be similar to each other, but not to tokens of other types – the example is simple enough not to require a hierarchy of types. 'Bite', 'flee', 'stroke', 'lick', and 'cause' are *relations*. *First-order* relations are those which have only objects as arguments, i.e., 'bite', 'flee', 'stroke', and 'lick'. Relations that have other relations as arguments are *higher-order* relations. 'Cause' is the only higher-order relation in these examples. None of these relations are considered to be similar to any other. Each argument position of a relation corresponds to a *role* and the argument is the *filler* of the role. In "Fido bit John" (i.e., bite(Fido,John)), Fido is the filler of the *agent* role and John is the filler of the *object* role. Objects, relation names, and roles all can have *features* or *attributes*, e.g., is-a-person is a feature of Jane, John, and Fred.

This work does not tackle the problem of how visual or natural language input is converted into propositional form, which is a difficult problem in its own right. As with ARCS and MAC/FAC, the propositional form of the stories is the starting point. The propositional form of the episodes in Table 4 is straightforward, e.g., the propositions corresponding to the probe are:

```
p1  =  bite(Spot, Jane)
p2  =  flee(Jane, Spot)
p3  =  cause(p1, p2)
```

## 4.1 Aspects of similarity

I have identified six aspects of the similarities between the probe and the memorized episodes. The distinctions among these aspects relate to different treatment by SME (Structure Mapping Engine) [Falkenhainer, Forbus and Gentner 1989] or ARCS (Analog Retrieval by Constraint Satisfaction)

[Thagard et al. 1990] or different effects on HRR dot-product similarity scores.

**OA**: Object attribute similarity – the presence of similar objects in each episode, e.g., people, dogs.

**FOR**: First-order relation similarity – the presence of similar first-order relations, e.g., bite, flee.

**HOR**: Higher-order relation similarity – the presence of similar higher-order relations, i.e., cause.

**RFB**: role-filler binding similarity – similar objects are bound to similar roles, e.g., a dog in the bite agent role. This only applies to bindings where the filler is an object, not another relation.

**HOS**: Higher-order relation structure similarity – similar relations are related in the same way, e.g., bite causes flee.

**OLI**: Object-level isomorphism – the presence of a consistent one-to-one mapping between the objects in the two episodes. This aspect is only important when objects fill multiple roles.

Object attribute similarity (OA), first-order relation similarity (FOR), and higher-order relation similarity (HOR) all depend on "surface" features – just the attributes of objects and relations. role-filler bindings similarity (RFB) depends on "local" structural features, while higher-order relation structure (HOS) and object-level isomorphism (OLI) depend upon the entire structural arrangement. I distinguish objects as fillers from propositions as fillers because the SME and ARCS treat objects differently to propositions.

Structural similarity (isomorphism) is split into two aspects – higher-order structure (HOS) and object-level isomorphism (OLI). It makes sense to treat the mapping of objects differently from the mapping of relations. The absence of similar objects does not at all diminish the strength of an analogy, and consequently object-level isomorphism does not require mapped objects to be similar. However, if two situations do not share similar relations, any analogy that can be constructed is likely to be abstract, difficult to find, and tenuous. Consequently, two episodes are similar in higher-order structure (HOS) if their higher-order structures are isomorphic and the corresponding predicates are similar. In SME objects mapped to each other can be dissimilar, but mapped relations must be identical. ARCS does allow mapping of non-identical relations, but only if they have the same number of arguments. The third structural aspect of similarity, role-filler binding (RFB) similarity is included because it has a large effect on HRR similarity but is ignored by SME and ARCS (although Markman *et al*[1993]) did note the stronger effect of "matches-in-place" versus "matches-out-of-place" on reminding and reasoning in human subjects).

In general, the rating of various aspects of similarity is best made on a continuous scale. However, the examples used here are so simple that a binary classification (✓or ×) suffices. The exception is **E**5, which gets half a mark for role-filler binding similarity (RFB). The fillers of the two bite roles are similar to those of the probe (both have a dog biting a human), but the fillers of the two flee roles are not.

## 4.2 Classifications of similarity

The final column in Table 4 classifies the relationship between each episode and the probe using Gentner, Rattermann and Forbus's [1993] types of similarity, which have the following relationships to the presence of the various aspects of similarity:

**LS** (Literal Similarity): all aspects.

**AN** (Analogy, also called True Analogy): FOR, HOR, HOS, and OLI, but not OA or not RFB.

**SS** (Surface Similarity, also called Mere Appearance): OA, FOR, and HOR, but no structural aspects.

**FA** (False Analogy[9]): FOR and HOR only.

**OO** (Only Objects similarity): OA only.

The superscripts on the types are for later identification and indicate how the episode differs from the probe. **E**2 ($AN^{cm}$) is called a *cross-mapped* analogy because it involves the same types of objects as the probe, but the types of corresponding objects do not match – the types are switched around. $SS^{\times I}$ indicates surface similarity without object-level isomorphism, $SS^{-H}$ indicates surface similarity with missing higher-order relations, and $SS^{\times H}$ indicates surface similarity with mismatching higher-order structure.

## 4.3 Ratings of the episodes

The first stages of MAC/FAC and ARCS only inspect surface features (OA, FOR, and HOR). The first stage of MAC/FAC (i.e., MAC, the "Many Are Called" stage) uses a vector representation of the surface features. Each location in the vector corresponds to a surface feature of an object, relation or function, and the value in the location is the number of times the feature occurs in the structure. The first-stage estimate of the similarity between two structures is the dot-product of their feature-count vectors. A threshold is used to select potential analogies. MAC would give the literally similar episode **E**1 (LS), the cross mapped analogy **E**2 ($AN^{cm}$), and the superficially similar episodes without higher-order similarity **E**6 ($SS^{\times H}$) or object-level isomorphism **E**5 ($SS^{\times I}$) equal and highest scores, because they all share the same number of object and predicate features with the probe. The first stage of ARCS is less selective than that of MAC/FAC. It selects all episodes that share a relation with the probe, ignoring ubiquitous relations such as "cause" and "if". It would select all episodes involving "bite" or "flee", i.e., **E**1-**E**9.

The second stages of MAC/FAC (i.e., SME) and ARCS use very similar rules: mapped relations must match, all the arguments of mapped relations must be mapped consistently, and mapping of objects must be one-to-one. One minor difference is that SME requires exact matches of relations, while ARCS allows mapping of similar relations. This is only a minor difference because similar relations can be decomposed into identical and dissimilar components. SME has two modes – literal similarity mode, and analogy mode. In literal similarity mode, mappings between similar objects are favored over those between dissimilar objects, which gives literal similarity a higher score than analogy ("literal similarity" is analogy plus object attribute matches). Where it is relevant, I assume that SME is operating in literal similarity mode.

I would expect the second stages of both MAC/FAC and ARCS to detect structural correspondences between each episode and the probe and give the literally similar and analogous episodes the highest rankings, i.e., LS > AN > (SS, FA, OO).

A simplified view of the overall similarity scores from fast and slow similarity estimators is shown in Table 5. There are four conditions – the two structures being compared can be similar in structure and/or in object attributes. In all four conditions, the structures are assumed to involve similar relations – only structural and object attribute similarities are varied. Gentner and Forbus's names for these classes are shown in parentheses. Ideally, the responses to the mixed conditions should be flexible, and controlled by which aspects of similarity are currently considered important. Only their relative values of the scores are important, the absolute values do not matter. The scores from the MAC similarity estimator (the first stage of MAC/FAC) are shown in Table 5(a). In the following subsections, I describe the HRRs for the episodes and how they can be used to compute fast similarity estimates that are more like ratings from SME and ARCS shown in Table 5(b).

[9] Gentner sometimes uses "First-Order Relational similarity" (FOR) instead of False analogy, but I avoid this term as it uses "First-order" in a different way (to refer to the consideration of relation names).

| Structural Similarity | Object Attribute Similarity YES | NO |
|---|---|---|
| YES | (LS) High | (AN) Low |
| NO | (SS) High | (FA) Low |

(a) Scores from the fast MAC similarity.

| Structural Similarity | Object Attribute Similarity YES | NO |
|---|---|---|
| YES | (LS) High | (AN) $\updownarrow$Med-High |
| NO | (SS) $\updownarrow$Med-Low | (FA) Low |

(b) Scores from a slow similarity estimator, e.g., SME or ARCS.

Table 5:

| Base vectors | | Token vectors |
|---|---|---|
| **person** | **bite** | $\mathbf{jane} = \langle \mathbf{person} + \mathbf{id}_{jane} \rangle$ |
| **dog** | **flee** | $\mathbf{john} = \langle \mathbf{person} + \mathbf{id}_{john} \rangle$ |
| **cat** | **cause** | $\mathbf{fred} = \langle \mathbf{person} + \mathbf{id}_{fred} \rangle$ |
| **mouse** | **stroke** | $\mathbf{spot} = \langle \mathbf{dog} + \mathbf{id}_{spot} \rangle$ |
| | **lick** | $\mathbf{fido} = \langle \mathbf{dog} + \mathbf{id}_{fido} \rangle$ |
| $\mathbf{bite}_{agt}$ | $\mathbf{bite}_{obj}$ | $\mathbf{rover} = \langle \mathbf{dog} + \mathbf{id}_{rover} \rangle$ |
| $\mathbf{flee}_{agt}$ | $\mathbf{flee}_{from}$ | $\mathbf{felix} = \langle \mathbf{cat} + \mathbf{id}_{felix} \rangle$ |
| $\mathbf{cause}_{antc}$ | $\mathbf{cause}_{cnsq}$ | $\mathbf{mort} = \langle \mathbf{mouse} + \mathbf{id}_{mort} \rangle$ |
| $\mathbf{stroke}_{agt}$ | $\mathbf{stroke}_{obj}$ | |
| $\mathbf{lick}_{agt}$ | $\mathbf{lick}_{obj}$ | |

Table 6:

## 4.4 Estimating similarity by dot-products of HRRs

### 4.4.1 Experiment 1

The dot-product of two HRRs gives an estimate of the overall similarity of the structures represented. This estimate combines the similarity on many different aspects, both surface and structural. The HRR dot-product is only an estimate of analogical match for two reasons: (1) the exact value of the dot product depends on essentially unpredictable interactions among the distributed representations of base vectors, and (2) the expected value of the HRR dot-product is an imperfect measure of analogical similarity.

Experiment 1, described in this section, illustrates the ways in which the dot-product of ordinary HRRs reflects, and fails to reflect, the similarity of the underlying structures. The set of base vectors and tokens used in Experiments 1, 2 and 3 is shown in Table 6. All base and **id** vectors are randomly chosen with elements independently distributed as $\mathrm{N}(0, 1/n)$.

The HRR for the probe is constructed as follows, and the HRRs for the other episodes are constructed in the same way.

$$
\begin{aligned}
\mathbf{P}_{bite} &= \langle \mathbf{bite} + \mathbf{bite}_{agt} \circledast \mathbf{spot} + \mathbf{bite}_{obj} \circledast \mathbf{jane} \rangle \\
\mathbf{P}_{flee} &= \langle \mathbf{flee} + \mathbf{flee}_{agt} \circledast \mathbf{jane} + \mathbf{flee}_{from} \circledast \mathbf{spot} \rangle \\
\mathbf{P}_{objects} &= \langle \mathbf{jane} + \mathbf{spot} \rangle \\
\mathbf{P} &= \langle \mathbf{cause} + \mathbf{P}_{objects} + \mathbf{P}_{bite} + \mathbf{P}_{flee} + \mathbf{cause}_{antc} \circledast \mathbf{P}_{bite} + \mathbf{cause}_{cnsq} \circledast \mathbf{P}_{flee} \rangle
\end{aligned}
$$

Table 7 summarizes the dot-products between HRRs for the probe and the episodes **E**1-**E**11, over

**P**: Spot bit Jane, causing Jane to flee from Spot.

| Episodes in long-term memory: | | Aspects of similarity OA | FOR | HOR | RFB | HOS | OLI | Type | Dot-products Avg | Sd |
|---|---|---|---|---|---|---|---|---|---|---|
| **E**1: | Fido bit John, causing John to flee from Fido. | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | LS | 0.70 | 0.016 |
| **E**2: | Fred bit Rover, causing Rover to flee from Fred. | ✓ | ✓ | ✓ | × | ✓ | ✓ | $AN^{cm}$ | 0.47 | 0.022 |
| **E**3: | Felix bit Mort, causing Mort to flee from Felix. | × | ✓ | ✓ | × | ✓ | ✓ | $AN_1$ | 0.39 | 0.024 |
| **E**4: | Mort bit Felix, causing Felix to flee from Mort. | × | ✓ | ✓ | × | ✓ | ✓ | $AN_2$ | 0.39 | 0.024 |
| **E**5: | Rover bit Fred, causing Rover to flee from Fred. | ✓ | ✓ | ✓ | $\frac{1}{2}$ | ✓ | × | $SS^{\times I}$ | 0.58 | 0.019 |
| **E**6: | John fled from Fido, causing Fido to bite John. | ✓ | ✓ | ✓ | ✓ | × | ✓ | $SS^{\times H}$ | 0.47 | 0.018 |
| **E**7: | Mort bit Felix, causing Mort to flee from Felix. | × | ✓ | ✓ | × | ✓ | × | $FA^{\times I}$ | 0.39 | 0.024 |
| **E**8: | Mort fled from Felix, causing Felix to bite Mort. | × | ✓ | ✓ | × | × | ✓ | $FA^{\times H}$ | 0.28 | 0.025 |
| **E**9: | Fido bit John, John fled from Fido. | ✓ | ✓ | × | ✓ | × | ✓ | $SS^{-H}$ | 0.43 | 0.019 |
| **E**10: | Fred stroked Rover, causing Rover to lick Fred. | ✓ | × | ✓ | × | × | × | $OO_1$ | 0.25 | 0.024 |
| **E**11: | Fred stroked Rover, Rover licked Fred. | ✓ | × | × | × | × | × | $OO_2$ | 0.12 | 0.023 |

Table 7:

100 trials, each with a different choice of random base vectors. The vector dimension used was 2048. In 94 out of 100 runs, the ranking of the HRR dot-products was consistent with the following order (where the ordering within the parenthesis varies):

$$LS > SS^{\times I} > (AN^{cm}, SS^{\times H}) > (FA^{\times I}, AN_1, AN_2, SS^{-H}) > (FA^{\times H}, OO_1) > OO_2$$

which can be summarized as LS > SS. This is also consistent with the order of the average dot-products. The order violations in individual runs are due to noise in the dot-products. The variance of the noise decreases as the vector dimension increases. When the experiment was rerun with vector dimension 4096 there was only one violation of this order out of 100 runs.

These results illustrate two important properties of the HRR dot-product estimate of similarity. The first is that the HRR dot-product is more sensitive to structural similarity when the objects involved are similar. The second is that the HRR dot-product is very sensitive to having similar roles filled by similar entities.

### 4.4.2 Conditions for structural similarity to affect the HRR dot-product

The HRR dot-product (as in Experiment 1) will give a high score to an analogous pair of episodes if two conditions are satisfied:

1. The corresponding pairs of objects are similar, i.e., they are of the same or similar types.
2. The objects within each episode are of distinct types.

The episodes **E**1 (LS), **E**5 ($SS^{\times I}$) and **E**6 ($SS^{\times H}$) satisfy these two conditions with respect to the probe **P**, and the HRR dot-product ranks them correctly. Each of these episodes involves one dog and one person, as does the probe. The literally similar episode **E**1, which is analogous to the probe, receives a higher score than the two superficially similar episodes **E**5 and **E**6.

The requirement for corresponding pairs of objects to be similar is illustrated by the low score given to the cross-mapped analogy **E**2. Even though this episode is structurally isomorphic to the probe it does not score higher than the two superficially similar episodes **E**5 ($SS^{\times I}$) and **E**6 ($SS^{\times H}$).

The requirement of object attribute similarity for the detection of structural similarity is illustrated by the relative scores of **E**3 ($AN_1$) and **E**7 ($FA^{\times I}$). These two episodes do not share object attributes with the probe, and receive approximately equal scores despite their different degrees of structural similarity.

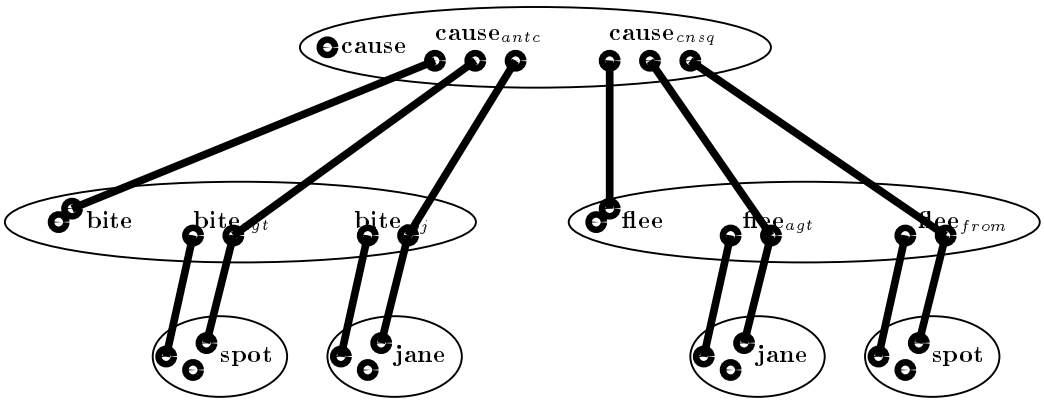



Figure 5: The HRR for the probe is the weighted sum of binding chains.

#### 4.4.3 Why the HRR dot-product reflects structural similarity

The HRR for an episode can be expanded to a weighted sum of vectors and convolution products of vectors. This is illustrated for the probe in Figure 5 (weights are not shown.) The vertically connected dots are the *binding chains*. For example, the rightmost binding chain is $\mathbf{cause}_{cnsq} \circledast \mathbf{flee}_{from} \circledast \mathbf{spot}$. The isolated dots represent that vector, unconvolved with any other. For the purposes of this discussion, these are binding chains involving just one vector. Many roles and fillers participate in more than one binding chain. For example, **spot** is in three bindings chains with lengths one, two and three. These convolution products contain one or more entities and correspond to the binding chains (hierarchical role-filler paths) in the episode. The weights come from normalizations performed during the construction of the HRR (and any relative component weights used). The binding chains provide a way of expressing structural features of an episode as superficial features of the HRR vector. Consideration of the binding chains shows both why the HRR dot-product reflects structural similarity under some conditions and why it is an unreliable indicator when those conditions are not satisfied. Two binding chains (convolution products) are similar if the entities in them are similar. Consequently, the HRRs for two episodes are similar if they have similar binding chains. Two episodes that are structurally similar and have similar fillers in similar roles will have similar binding chains. However, structurally similar episodes that do not have similar fillers in similar roles will not have similar binding chains (e.g., cross-mapped analogies). And episodes that have many similar fillers in similar roles but are structurally dissimilar may still have many similar binding chains (e.g., superficially similar episodes).

The vectors and binding chains which cause sensitivity to the six aspects of similarity are as follows:

**OA** (Object Attributes): Fillers in the top-level relation (e.g., $\mathbf{P}_{objects}$ in $\mathbf{P}$).

**FOR** (First-order relation names): Relation names in the lower-level relations, and those relations in the top-level relation (e.g., **bite** in $\mathbf{P}_{bite}$ and **flee** in $\mathbf{P}_{flee}$, and $\mathbf{P}_{bite}$ and $\mathbf{P}_{flee}$ in $\mathbf{P}$).

**HOR** (Higher-order relation names): Relation names in the higher-level relations (e.g., **cause** in $\mathbf{P}$).

**RFB** (Role-filler bindings): First-order role-filler bindings, and those relations in the top-level relation (e.g., bindings in $\mathbf{P}_{bite}$ and $\mathbf{P}_{flee}$, and $\mathbf{P}_{bite}$ and $\mathbf{P}_{flee}$ in $\mathbf{P}$).

**HOS** (Higher-order structure): Relation names in lower-level relations, and the role-filler bindings those relations participate in (e.g., **bite** in $\mathbf{P}_{bite}$ and **flee** in $\mathbf{P}_{flee}$, and the bindings $\mathbf{cause}_{antc} \circledast \mathbf{P}_{bite}$ and $\mathbf{cause}_{cnsq} \circledast \mathbf{P}_{flee}$ in $\mathbf{P}$).

**OLI** (Object-level isomorphism): HRRs are not directly sensitive to this, though, in many situations OLI is correlated with role-filler binding similarity (RFB).

SME constrains corresponding higher-order relations to be identical. If this is a valid constraint, or even if just a high degree of similarity is required, then the detection of higher-order relational similarity is simpler than the detection of object-level isomorphism. Under this definition, higher-order relational structure is expressed as a superficial feature in a HRR vector – increasing structural similarity (while keeping the degree surface of surface similarity constant) will always increase the HRR dot-product. However, the object-level filler structure is not expressed as a superficial feature in the HRR vector – two episodes may have completely isomorphic object-level filler structure and low HRR dot-product similarity.

Compared to the similarity measures embodied in SME and ARCS, HRR dot-products are especially sensitive to the presence of similar role-filler bindings. In HRRs, binding similarity is the only mechanism by which structural similarity is detected. SME and ARCS use different (more rigorous) methods to detect structural isomorphism, and are more relaxed about binding similarity. However, they do take the similarity of mapped entities into account, and this is closely related to role-filler binding similarity. SME insists that mapped relations should be identical (in name), but does not take account of the similarity of mapped objects unless it is operating in "Literal similarity" mode. ARCS has the soft constraint that all mapped entities, objects and relation names, should be similar.

A set of binding chains is not an unambiguous description of a structure – there are distinct structures that have identical sets of binding chains. The simplest examples of this are episodes with two instances of the same proposition, and no higher-order propositions. For example "Spot bit Jane, Fido bit Fred" has the same set of binding chains as "Spot bit Fred, Fido bit Jane".[10] If this turns out to be a problem, one could include filler-filler bindings (i.e., $\mathbf{spot} \circledast \mathbf{jane}$ for "Spot bit Jane") in order to better indicate which fillers belong to the same relation. The associativity and commutativity of convolution is another potential source of ambiguity. These properties result in the convolution product representation for binding chains being order-insensitive. This means that some binding chains arising from nested propositions are the same regardless of the nesting order of the propositions. If this turns out to be a serious problem, there are non-commutative versions of convolution which can be used [Plate 1995; Plate 1994].

### 4.4.4 HRRs and multiple occurrences of objects

HRRs do not encode multiple occurrences of a single object as an explicit feature. Each occurrence is treated as a separate object – no links are made among multiple occurrences. This has several consequences for the HRR dot-product as a measure of analogical similarity. One is that HRRs are insensitive to object-level isomorphism (OLI) in the absence of object attribute similarity (OA). Another is that lack of object-level isomorphism is not detected if there are multiple objects of the same type. The HRR dot-product of the probe **P** and an episode will be high as long as the episode involves a dog biting a person, causing a person to run away from a dog. It does not matter whether the first person is the same as the second person, or the first dog the same as the second. For example, "Fido bit John, causing Fred to run away from Rover" would score just as highly as **E**1. HRRs dot-products are only sensitive to object-level isomorphism (OLI) through virtue of sensitivity to the presence of similar role-filler bindings (RFB). In Section 4.5, I describe how links can be made among the multiple occurrences of an object.

[10] If chunks for "Spot bit Jane" and "Fido bit Fred" are stored separately, these episodes can still be correctly decoded.

### 4.4.5 Weighting different aspects of the HRR

If certain aspects of the structures should have more influence than others on the HRR similarity scores, then these aspects can be given a higher relative weight in the HRR. For example, if it were desired that objects should have a very large influence on HRR similarity scores, the component $\mathbf{P}_{objects}$ could be scaled up before adding it to the other components of $\mathbf{P}$. Changes in weighting only need to be done in one HRR of the comparison to produce an effect. This is computationally convenient – it means that the salience of different aspects of similarity can be changed by just reweighting components of the probe.

A rerun of Experiment 1 with the objects ($\mathbf{P}_{objects}$) in the probe given five times higher weight, i.e.,

$$\mathbf{P}' = \langle \mathbf{cause} + 5 \cdot \mathbf{P}_{objects} + \mathbf{P}_{bite} + \mathbf{P}_{flee} + \mathbf{cause}_{antc} \circledast \mathbf{P}_{bite} + \mathbf{cause}_{cnsq} \circledast \mathbf{P}_{flee} \rangle,$$

with unchanged HRRs for each episode, satisfied the following order, with one violation out of 100:

$$\mathrm{LS} > (\mathrm{SS}^{\times \mathrm{I}}, \mathrm{SS}^{-\mathrm{H}}, \mathrm{SS}^{\times \mathrm{H}}, \mathrm{AN}^{\mathrm{cm}}) > (\mathrm{OO}_1, \mathrm{OO}_2) > (\mathrm{AN}_1, \mathrm{AN}_2, \mathrm{FA}^{\times \mathrm{I}}) > \mathrm{FA}^{\times \mathrm{H}})$$

In this ranking the episodes which share object attributes with the probe score higher than those which do not.

In the HRRs described in this paper, some relative weighting is accomplished by the normalizations. The components $\mathbf{P}_{bite}$, $\mathbf{P}_{flee}$, and $\mathbf{P}_{objects}$ are all normalized (by a factor of approximately $1/\sqrt{3}$, $1/\sqrt{3}$, and $1/\sqrt{2}$ respectively) so that they have equal weight in the final HRR. Thus the predicate names **bite** and **flee** have less weight in $\mathbf{P}$ than **cause**.

### 4.4.6 Covariance of HRR dot-products

If two episodes in memory are similar, their dot-products with the probe will tend to be correlated. This makes ordering of similarity more reliable than might be deduced by looking at the sample variances (shown in Table 7). For example, the difference between $\mathbf{P} \cdot \mathbf{E}1$ and $\mathbf{P} \cdot \mathbf{E}6$ (in each run) has an average value of 0.22 and a sample standard deviation of 0.017. If these two dot-products were uncorrelated the standard deviation of the difference would be around 0.024 (from summing the variances of $\mathbf{P} \cdot \mathbf{E}1$ and $\mathbf{P} \cdot \mathbf{E}6$). The correlation coefficient between $\mathbf{P} \cdot \mathbf{E}1$ and $\mathbf{P} \cdot \mathbf{E}6$ was 0.46 (which is significant beyond 0.2%). The correlation comes from common terms in the expansion of the dot-product, e.g., both $\mathbf{E}1$ and $\mathbf{E}6$ have $\mathbf{cause} + \langle \mathbf{john} + \mathbf{fido} \rangle$ as a component, thus $(\mathbf{cause} + \langle \mathbf{john} + \mathbf{fido} \rangle) \cdot (\mathbf{cause} + \langle \mathbf{john} + \mathbf{fido} \rangle)$ contributes to both $\mathbf{P} \cdot \mathbf{E}1$ and $\mathbf{P} \cdot \mathbf{E}6$. This effect is much larger when base vectors and chunks are not normalized, but are chosen to have an expected length of one. The effect is larger because individual vectors contribute correlated values (e.g., $\mathbf{cause} \cdot \mathbf{cause}$) to the final dot-products, whereas when all entities are normalized these contributed values are constant (because $\mathbf{cause} \cdot \mathbf{cause} = 1$). An experiment without normalization yielded a correlation coefficient of 0.75 for $\mathbf{P} \cdot \mathbf{E}1$ and $\mathbf{P} \cdot \mathbf{E}6$. Standard deviations of these dot-products (and differences) were 50 to 150% higher than with normalization.

### 4.4.7 Properties of the HRR dot-product as a measure of similarity

Under certain conditions the HRR dot-product provides a good indication of overall similarity. However, it has two weaknesses:

1. It overestimates when structural similarity is low and there are many similar role-filler bindings.
2. It underestimates when structural similarity is high and there are few similar role-filler bindings.

These weaknesses are in part due to HRRs not encoding the multiple occurrence of a single entity as an explicit surface feature. The occurrence of a single entity in a multiple roles is what makes structural alignment difficult. If each entity occurs only once then the combinatoric complexity of exact structural alignment is much lower: the problem is one of tree matching rather than graph matching.

This reliance on the presence of similar role-filler bindings as an indicator of structural similarity is the underlying reason for the failure of the HRR dot-product to detect structural similarity when corresponding fillers are not similar. The next section presents an enhancement of HRRs intended to address this problem. This enhancement involves special treatment for objects which fill multiple roles and improves performance on cross-mapped analogies and on analogies without similar objects

## 4.5 Contextualized HRRs

Dot-product comparisons of HRRs are not sensitive to object-level isomorphism, especially in the absence of similarity among objects. This is because the way in which objects fill multiple roles is not expressed as a surface feature in HRRs. Consequently, the analogical episodes **E**2 ($\mathrm{AN}^{\mathrm{cm}}$), **E**3 ($\mathrm{AN}_1$), and **E**4 ($\mathrm{AN}_2$) are given lower scores than the superficially similar episode **E**5 ($\mathrm{SS}^{\times \mathrm{I}}$).

We can force role structure to become a surface feature (and thus allow the degree of role alignment to influence the dot-product comparison) by "contextualizing" the representations of fillers. Contextualization involves incorporating information about what other roles an object fills in the representation of a filler. This is like conceiving of Spot (in the probe) as an entity which bites (a biter) and an entity which is fled from (a "fledfrom"). Contextualization can be implemented in HRRs by blending the representation for a object with a representation for the typical fillers of the other roles the object fills. This can be viewed as abstracting the representation of a filler: its representation involves both its identity and the contexts it appears in.

One possible representation for the "biter" feature is:

$$\mathbf{bite}_{typagt} = \mathbf{bite} * \mathbf{bite}^*_{agt},$$

where $\mathbf{bite}^*_{agt}$ is the approximate inverse of $\mathbf{bite}_{agt}$. This features marks entities as being a typical filler of the bite agent role. This representation for $\mathbf{bite}_{typagt}$ is attractive because it is a component of the result of decoding the agent of any "bite" proposition. The agent of a proposition is decoded by convolving with the approximate inverse of the appropriate agent vector as follows:

$$\begin{aligned}(\mathbf{bite} + \mathbf{bite}_{agt} \circledast \mathbf{spot} + \mathbf{bite}_{obj} \circledast \mathbf{jane}) \circledast \mathbf{bite}^*_{agt} \\ \approx \quad \mathbf{spot} + \underline{\mathbf{bite} \circledast \mathbf{bite}^*_{agt}} + \mathbf{bite}_{obj} \circledast \mathbf{jane} \circledast \mathbf{bite}^*_{agt}\end{aligned}$$

The underlined term is always present when an agent is extracted from a bite proposition, but would usually be treated as noise. The attractiveness of this representation for typical fillers is in its potential for developing representations of entities that contain information about the types of roles and relationships the entity is commonly involved in. If the long-term memory representation for Spot is somewhat malleable, and if Spot is decoded as the agent in many bite propositions, then the representation for Spot might over time incorporate this component which represents a typical biter. However, the implementation of this is outside the scope of this paper.

In ordinary HRRs the filler alone is convolved with the role. In contextualized HRRs a blend of the filler and its context is convolved with the role. The representation for the context of object in a role is the typical fillers of the *other* roles the object fills. In the **bite** relation the context for Spot is represented by $\mathbf{flee}_{typfrom}$, and in the **flee** relation the context for Spot is represented by $\mathbf{bite}_{typagt}$. The degree of contextualization is governed by the mixing proportion $\kappa_o$ (object) and $\kappa_c$ (context). The contextualized HRR for the probe is constructed as follows:

$$
\begin{aligned}
\mathbf{P}_{bite} &= \langle \mathbf{bite} + \mathbf{bite}_{agt} \circledast (\kappa_o \mathbf{spot} + \kappa_c \mathbf{flee}_{typfrom}) \\
&\quad + \mathbf{bite}_{obj} \circledast (\kappa_o \mathbf{jane} + \kappa_c \mathbf{flee}_{typagt}) \rangle \\
\mathbf{P}_{flee} &= \langle \mathbf{flee} + \mathbf{flee}_{agt} \circledast (\kappa_o \mathbf{jane} + \kappa_c \mathbf{bite}_{typobj}) \\
&\quad + \mathbf{flee}_{from} \circledast (\kappa_o \mathbf{spot} + \kappa_c \mathbf{bite}_{typagt}) \rangle \\
\mathbf{P}_{objects} &= \langle \mathbf{jane} + \mathbf{spot} \rangle \\
\mathbf{P}_{probe} &= \langle \mathbf{cause} + \mathbf{P}_{objects} + \mathbf{P}_{bite} + \mathbf{P}_{flee} + \mathbf{cause}_{antc} \circledast \mathbf{P}_{bite} + \mathbf{cause}_{cnsq} \circledast \mathbf{P}_{flee} \rangle
\end{aligned}
$$

The degree of contextualization controls the degree to which the pattern of fillers is expressed as a surface feature, and hence the controls the degree to which role alignment influences the dot-product.

### 4.5.1 Flexible salience of object-level isomorphism

Human judgements of similarity are very flexible – the salience of different aspects of similarity can be changed by context or command. The degree to which role alignment affects the HRR dot-product can be adjusted by changing the degree of contextualization in just one episode of a pair. Hence, the items in memory can be encoded with a fixed $\kappa$ values ($\kappa_o^m$ and $\kappa_c^m$) and the salience of role alignment can be changed by altering the degree of contextualization in the probe ($\kappa_o^p$ and $\kappa_c^p$). This is fortunate as it would be impractical to recode all items in memory in order to alter the salience of role alignment in a particular comparison.

## 4.6 Estimating similarity by dot-products of contextualized HRRs

Two experiments were performed with contextualized HRRs, with the same episodes as used in Experiment 1. In Experiment 2 the probe was non-contextualized ($\kappa_o^p = 1, \kappa_c^p = 0$), and in Experiment 3 contextualization was used ($\kappa_o^p = 1/\sqrt{2}, \kappa_c^p = 1/\sqrt{2}$). The episodes in memory were encoded with the same degree of contextualization ($\kappa_o^m = 1/\sqrt{2}, \kappa_c^m = 1/\sqrt{2}$) for both experiments. As before, each set of comparisons was run 100 times, and the vector dimension was 2048. The results are listed in Table 9. Only the means of the dot-products are shown – the standard deviations were very similar to those in Experiment 1.

### 4.6.1 Experiments 2 and 3

The dot-products in Experiment 2 were consistent (in 95 out of 100 runs) with the same order as given for Experiment 1:

$$\mathrm{LS} > \mathrm{SS}^{\times \mathrm{I}} > (\mathrm{AN}^{\mathrm{cm}}, \mathrm{SS}^{\times \mathrm{H}}) > (\mathrm{FA}^{\times \mathrm{I}}, \mathrm{AN}_1, \mathrm{AN}_2, \mathrm{SS}^{-\mathrm{H}}) > (\mathrm{FA}^{\times \mathrm{H}}, \mathrm{OO}_1) > \mathrm{OO}_2$$

The dot-products in Experiment 3 were consistent (in all 100 runs) with an ordering which ranks analogical episodes as strictly more similar than non-analogical ones:

$$\mathrm{LS} > \mathrm{AN}^{\mathrm{cm}} > (\mathrm{AN}_2, \mathrm{AN}_1) > (\mathrm{SS}^{\times \mathrm{I}}, \mathrm{SS}^{\times \mathrm{H}}, \mathrm{SS}^{-\mathrm{H}}) > (\mathrm{FA}^{\times \mathrm{I}}, \mathrm{FA}^{\times \mathrm{H}}) > \mathrm{OO}_1 > \mathrm{OO}_2$$

### 4.6.2 Discussion of results

When there is no contextualization in probe ($\kappa_o^p = 1, \kappa_c^p = 0$) the ordering of dot-products is the same as for Experiment 1. With contextualization, the analogous episodes **E**2 ($\mathrm{AN}^{\mathrm{cm}}$), **E**3 ($\mathrm{AN}_1$), and **E**4 ($\mathrm{AN}_2$) receive a higher score than the superficially similar episode **E**5 ($\mathrm{SS}^{\times \mathrm{I}}$).

**P**: Spot bit Jane, causing Jane to flee from Spot.

| Episodes in long-term memory: | | Aspects of similarity OA | FOR | HOR | RFB | HOS | OLI | Type | Avg. dot-prods Expt2 | Expt3 |
|---|---|---|---|---|---|---|---|---|---|---|
| **E**1: | Fido bit John, causing John to flee from Fido. | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | LS | 0.63 | 0.81 |
| **E**2: | Fred bit Rover, causing Rover to flee from Fred. | ✓ | ✓ | ✓ | × | ✓ | ✓ | $AN^{cm}$ | 0.47 | 0.69 |
| **E**3: | Felix bit Mort, causing Mort to flee from Felix. | × | ✓ | ✓ | × | ✓ | ✓ | $AN_1$ | 0.39 | 0.61 |
| **E**4: | Mort bit Felix, causing Felix to flee from Mort. | × | ✓ | ✓ | × | ✓ | ✓ | $AN_2$ | 0.39 | 0.61 |
| **E**5: | Rover bit Fred, causing Rover to flee from Fred. | ✓ | ✓ | ✓ | $\frac{1}{2}$ | ✓ | × | $SS^{\times I}$ | 0.55 | 0.53 |
| **E**6: | John fled from Fido, causing Fido to bite John. | ✓ | ✓ | ✓ | ✓ | × | ✓ | $SS^{\times H}$ | 0.44 | 0.53 |
| **E**7: | Mort bit Felix, causing Mort to flee from Felix. | × | ✓ | ✓ | × | ✓ | × | $FA^{\times I}$ | 0.39 | 0.39 |
| **E**8: | Mort fled from Felix, causing Felix to bite Mort. | × | ✓ | ✓ | × | × | ✓ | $FA^{\times H}$ | 0.27 | 0.39 |
| **E**9: | Fido bit John, John fled from Fido. | ✓ | ✓ | × | ✓ | × | ✓ | $SS^{-H}$ | 0.38 | 0.51 |
| **E**10: | Fred stroked Rover, causing Rover to lick Fred. | ✓ | × | ✓ | × | × | × | $OO_1$ | 0.25 | 0.25 |
| **E**11: | Fred stroked Rover, Rover licked Fred. | ✓ | × | × | × | × | × | $OO_2$ | 0.12 | 0.12 |

Table 8: Results from Experiments 2 and 3.

Table 9:

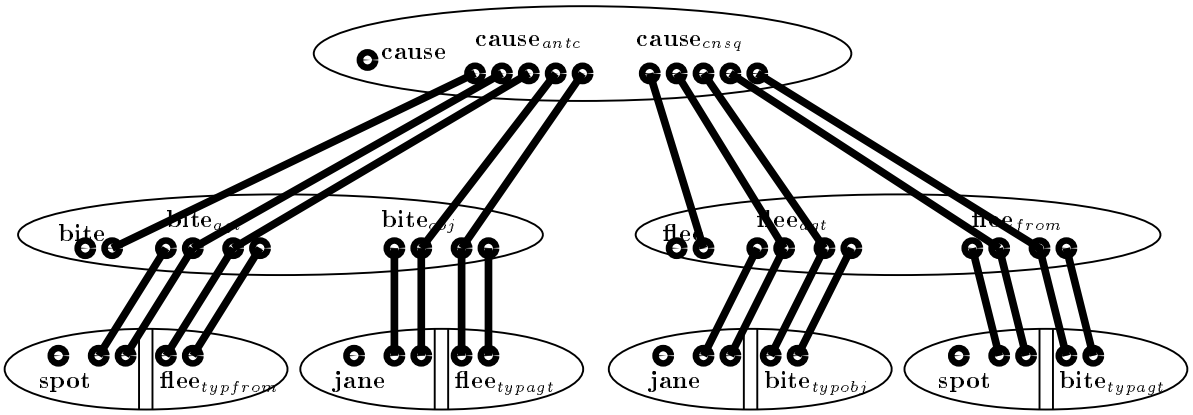



Figure 6: The binding chains in the contextualized HRR for the probe.

The example here shows the use of contextualization for objects, but it can also be used for relations. The example is not complex enough to have a relation appearing in multiple roles. For example, if "Jane fled from Spot" filled a role in another relation in addition to filling the cause antecedent role, then **cause**$_{typcnsq}$ would be the context used in that other relation.

### 4.6.3 Extra binding chains introduced by contextualization

Contextualization nearly doubles the number of binding chains in the HRR for an episode (and would increase their number more if some entities were involved in more than two relations). Figure 6 shows the binding chains in the contextualized HRR for the probe. The extra binding chains encode the patterns in which objects fill multiple roles.

### 4.6.4 Limits of contextualization

Contextualization does *not* cause all relational structure to be expressed as surface features in the HRR vector. It only suffices to distinguish analogical from non-analogical structures when no two entities fill the same set of roles. Sometimes, the distinguishing context for an object is more than the other roles that the object fills. Consider the situation where two boys are bitten by two dogs,

and each flees from the dog which did not bite him. With contextualization as described above it is impossible to distinguish this from the situation where each boy flees from the dog that did bite him. The point is that the structural context required to distinguish objects is more than the set of roles the objects fill. This example with two boys and two dogs can be written as follows (where $p$ is the bite relation and $q$ is the flee relation):

$$\begin{aligned} S_1 &= p(x,y), p(w,z), q(z,x), q(y,w) \\ S_2 &= p(a,b), p(c,d), q(d,a), q(b,c) \\ S_3 &= p(a,b), p(c,d), q(b,a), q(d,c) \end{aligned}$$

$S_2$ is analogous to $S_1$, but $S_3$ is not. However, if objects are ignored, they all have the same contextualized representation (where $tf_{q_1}$ is the typical filler of the first role of $q$ etc):

$$p(tf_{q_2}, tf_{q_1}), p(tf_{q_2}, tf_{q_1}), q(tf_{p_2}, tf_{p_1}), q(tf_{p_2}, tf_{p_1})$$

One could try to get around these limits on contextualization by incorporating more information into filler contexts, but the scheme quickly loses its simplicity. Problems with circular definitions arise – if other fillers are taken into account then the contextualized representation of $x$ in $p(x,y)$ involves the contextualized representation of $y$, which involves the contextualized representation of $x$. The solution probably lies in performing deeper contextualization only where necessary, for example, contextualizing John as "the person who fled from the dog which didn't bite him".

What is important is whether these limitations of contextualization ever manifest themselves in practical problems. Contextualization is more powerful than it appears at first sight – it fails only when there is a high degree of symmetry within as well as between episodes.

# 5 Interpretations of an analogy

To accurately evaluate the goodness of an analogy, or to use an analogy to help with reasoning, we need to work out which entities correspond to each other. The dot-product of HRRs does not provide this. However, corresponding entities can be easily computed from HRRs, by decoding the appropriate roles and fillers using the techniques described in Section 2.6.

Consider the probe **P** "Spot bit Jane, causing Jane to flee from Spot", and **E**1 "Fido bit John, causing John to flee from Fido." The entity corresponding to Jane (i.e., John) can be found in two steps:

1. Extract the roles Jane fills from the probe with the operation: $\langle \mathbf{P} \circledast \mathbf{jane}^* \rangle$ The roles which have positive dot-products with this result are:

| $\langle \mathbf{P} \circledast \mathbf{jane}^* \rangle$ | | |
|---|---|---|
| | $\mathbf{flee}_{agt}$ | 0.17 |
| | $\mathbf{cause}_{antc}$ | 0.16 |
| | $\mathbf{bite}_{agt}$ | 0.13 |
| | $\mathbf{cause}_{cnsq}$ | 0.10 |
| | $\mathbf{stroke}_{agt}$ | 0.01 |

2. Extract the fillers of those roles from **E**1 and compare (dot-product) with the entities in **E**1:

| | | | |
|---|---|---|---|
| $\langle \mathbf{E}1 \circledast (\mathbf{P} \circledast \mathbf{jane}^*)^* \rangle$ | **john** | 0.37 | ✓ |
| | **fido** | 0.05 | |
| $\langle \mathbf{E}2 \circledast (\mathbf{P} \circledast \mathbf{jane}^*)^* \rangle$ | **fred** | 0.25 | × |
| | **rover** | 0.10 | |
| $\langle \mathbf{E}3 \circledast (\mathbf{P} \circledast \mathbf{jane}^*)^* \rangle$ | **mort** | 0.14 | ✓ |
| | **felix** | 0.02 | |
| $\langle \mathbf{E}4 \circledast (\mathbf{P} \circledast \mathbf{jane}^*)^* \rangle$ | **felix** | 0.12 | ✓ |
| | **mort** | 0.07 | |
| $\langle \mathbf{E}5 \circledast (\mathbf{P} \circledast \mathbf{jane}^*)^* \rangle$ | **fred** | 0.26 | ✓ |
| | **rover** | 0.08 | |

(a)

| | | | |
|---|---|---|---|
| $\langle \mathbf{E}1 \circledast \mathbf{R}^* \rangle$ | **john** | 0.22 | ✓ |
| | **fido** | 0.11 | |
| $\langle \mathbf{E}2 \circledast \mathbf{R}^* \rangle$ | **rover** | 0.25 | ✓ |
| | **fred** | 0.08 | |
| $\langle \mathbf{E}3 \circledast \mathbf{R}^* \rangle$ | **mort** | 0.24 | ✓ |
| | **felix** | 0.10 | |
| $\langle \mathbf{E}4 \circledast \mathbf{R}^* \rangle$ | **felix** | 0.25 | ✓ |
| | **mort** | 0.10 | |
| $\langle \mathbf{E}5 \circledast \mathbf{R}^* \rangle$ | **fred** | 0.21 | ✓ |
| | **rover** | 0.17 | |

(b)

Table 10:

| | | |
|---|---|---|
| $\langle \mathbf{E}1 \circledast (\mathbf{P} \circledast \mathbf{jane}^*)^* \rangle$ | **john** | 0.37 |
| | **fido** | 0.05 |

The most similar entity is John, which is the entity in $\mathbf{E}1$ corresponding to Jane.

The extraction, without intermediate role clean-up, of the entities corresponding to Jane in $\mathbf{E}1 - \mathbf{E}5$ is shown in Table 10(a). Correct extractions are checkmarked. The correct answer is obtained in $\mathbf{E}1$ (LS), where corresponding objects are similar, and in $\mathbf{E}3$ ($\mathrm{AN}_1$) and $\mathbf{E}4$ ($\mathrm{AN}_2$), where there is no object similarity. This extraction process has a bias towards choosing similar entities as the corresponding ones. There is no correct answer for $\mathbf{E}5$ ($\mathrm{SS}^{\times \mathrm{I}}$), because there is no consistent mapping between $\mathbf{P}$ and $\mathbf{E}5$. However, Fred, (the entity more similar to Jane) is strongly indicated to be the one corresponding to Jane. The wrong answer is given for the cross-mapped analogy $\mathbf{E}2$, where again the more similar object is indicated to be the corresponding one.

Closer examination of the extraction process reveals both the reason for this bias and a way of eliminating it. Consider the expansion of $\mathbf{P} \circledast \mathbf{jane}^*$ (omitting weighting factors):

$$\mathbf{P} \circledast \mathbf{jane}^* = \ldots + \mathbf{bite}_{obj} + \ldots + \mathbf{flee}_{agt} + \ldots + \mathbf{cause} \circledast \mathbf{jane}^* + \ldots$$

The terms $\mathbf{bite}_{obj}$ and $\mathbf{flee}_{agt}$ are what we want – they come from $\mathbf{bite}_{obj} \circledast \mathbf{jane} \circledast \mathbf{jane}^*$ and $\mathbf{flee}_{agt} \circledast \mathbf{jane} \circledast \mathbf{jane}^*$). However, the other terms like $\mathbf{cause} \circledast \mathbf{jane}^*$ are the source of the bias. The approximate inverse[11] of $\mathbf{P} \circledast \mathbf{jane}^*$ is:

$$(\mathbf{P} \circledast \mathbf{jane}^*)^* = \ldots + \mathbf{bite}^*_{obj} + \ldots + \mathbf{flee}^*_{agt} + \ldots + \mathbf{cause}^* \circledast \mathbf{jane} + \ldots$$

When this is used to extract the fillers from $\mathbf{E}2$,

$$(\mathbf{P} \circledast \mathbf{jane}^*)^* \circledast \mathbf{E}2,$$

the result includes **fido** from $\mathbf{bite}_{obj} \circledast \mathbf{fido} \circledast \mathbf{bite}^*_{obj}$ and $\mathbf{flee}_{agt} \circledast \mathbf{fido} \circledast \mathbf{flee}^*_{agt}$. The result also includes a large component of **jane** from the many terms like $\mathbf{cause} \circledast \mathbf{cause}^* \circledast \mathbf{jane}$. The **jane** component is much larger than the **fido** component, and as **jane** is similar to **fred** the result ends up being more similar to **fred** than **fido**.

This bias can be eliminated by performing a special intermediate clean-up on the roles extracted by the operation $\mathbf{P} \circledast \mathbf{jane}^*$. This more complex clean-up should pass all positive role components and suppress negative role and non-role components like $\mathbf{cause} \circledast \mathbf{jane}^*$. For example, the cleaned-up version of $\mathbf{P} \circledast \mathbf{jane}^*$ should be:

$$\mathbf{R} = (0.17 * \mathbf{flee}_{agt} + 0.15 * \mathbf{bite}_{obj} + 0.13 * \mathbf{cause}_{antc} + 0.10 * \mathbf{cause}_{cnsq} + 0.01 * \mathbf{stroke}_{agt}),$$

[11] The approximate inverse obeys the axioms $(a + b)^* = a^* + b^*$ and $(a \circledast b^*)^* = a^* \circledast b$.

| SME | ARCS |
|---|---|
| **Clarity:** it is clear which objects in the two situations correspond to each other.<br>**Richness:** analogies with more correspondences are better.<br>**Abstractness:** correspondences among abstract entities are better:<br>• Attributes (one-place predicates) are not mapped (unless operating in literal similarity mode).<br>• Higher-order relations are more important.<br>**Systematicity:** consistent mappings are required. This involves:<br>• Presence of only one-to-one mappings.<br>• Structural consistency – a relation can only be in a mapping only if its arguments are consistently mapped as well. | **Semantic Similarity:** Mapped elements should be similar.<br>**Isomorphism:** Consistent mappings are required:<br>• Mappings should be one-to-one.<br>• Structural consistency – if a relation is mapped, then its arguments should be mapped consistently as well.<br>**Pragmatic centrality:** Either a particular correspondence holds or a particular element is involved in some correspondence. |

Table 11:

where the weights come from the dot-products of $\mathbf{P} \circledast \mathbf{jane}^*$ and the roles (listed in step 1 at the beginning of this section). This clean-up can be viewed as a less competitive version of the standard clean-up. The corresponding objects extracted using role clean-up are shown in Table 10(b). The process gives the correct answers for all four episodes where there is a consistent mapping, and gives an ambiguous answer for the episode which has no consistent mapping with the probe. The process is not sensitive to the weights used in the clean-up – similar answers are obtained if the roles are added together without weights (as long as just those roles which have a positive dot-product with $\mathbf{P} \circledast \mathbf{jane}^*$ are used), i.e,

$$\mathbf{R} = (\mathbf{flee}_{agt} + \mathbf{bite}_{obj} + \mathbf{cause}_{antc} + \mathbf{cause}_{cnsq} + \mathbf{stroke}_{agt}).$$

There are two problems with these techniques for deriving interpretations. One is that each pair in the mapping is extracted independently. This matters when there is more than one consistent mapping. For example, if we have two possible mappings $\{X \leftrightarrow A, Y \leftrightarrow B\}$ and $\{X \leftrightarrow B, Y \leftrightarrow A\}$, then the choice of mapping for $X$ should constrain the choice for $Y$. Another problem is that these techniques are unsatisfactory when two different objects have the same set of roles – ambiguous results can be produced.

# 6 Discussion

## 6.1 The criteria for good analogies in SME and ARCS

The criteria for good analogies in SME and ARCS are very similar and are listed in Table 11. SME and ARCS are intended to model analogical reasoning in people, so these criteria come from observations, experiments, and theories about how people judge and reason with analogies.

The degrees to which dot-products of ordinary and contextualized HRRs can be seen as implementing the criteria in Table 11 are as follows:

**Clarity:** No. However, an indication of clarity can be obtained by computing corresponding objects and seeing how clear the choices are.

**Richness:** Yes. More similar components means a higher dot-product.

**Abstractness:** Not with ordinary HRRs, to some degree with contextualized HRRs. Having similar attributes on corresponding objects does contribute to dot-product similarity, but can be made less important by using contextualization. Higher-order relations can be given more importance by using them larger weights for them.

**Semantic similarity:** Yes. Semantic similarity between corresponding entities increases the HRR dot-product. (This is the opposite of abstractness.)

**Systematicity (isomorphism):**

- Structural consistency: To some degree. Two instances of a higher-order relation will be more similar if their arguments are similar.
- One-to-one mappings: Not with ordinary HRRs, to some extent with contextualized HRRs.

**Pragmatic centrality:** No. However, if it were important that a particular object or relation be matched, it could be given a higher weight.

## 6.2 Flexibility of comparisons

Gentner and Markman [1993] and Goldstone, Medin and Gentner [1991] present convincing evidence that people are flexible in how they evaluate or use analogies. People can take different aspects of situations into account, depending on what is salient in the given context. In the HRR model all aspects are added together but flexibility is retained – different aspects in the representations can be weighted as appropriate. This requires only changing the representation of the probe; the representations of items in long-term memory can be fixed.

## 6.3 Features or predicates?

In HRRs, features and predicates have different representations. This creates the problem of having to decide which to use. The choice can dramatically affect the similarity of ordinary HRRs, but has a lesser effect on the similarity of contextualized HRRs.

In their commentary on the shape configuration experiments, Markman *et al* suggest Items 5 and 6 demonstrate that multiple relations can be used in the same comparisons ("small object above large object" is more similar to "small object above large object" than "large object above small object"). In the HRR encoding I used features rather than relations to represent the relative sizes, and this gave acceptable results. It would be possible to encode the relative sizes as a predicate "is larger than" (with roles "larger" and "smaller"), rather than as features of the objects. However, if this were done the dot-product comparisons of ordinary HRRs would not produce the correct answers for Item 6, because there would be no explicit relationship (binding) between "smaller" and "above". Using contextualization and a "is larger than" predicate would have a very similar effect to using the features "large" and "small", because contextualization would provide a "typical larger object" feature and a "typical smaller object" feature to be added to the shape features (circle, star,

etc). This suggests that there is something right about contextualization – if it lessens the artificial distinction between features and predicates

This issue is related to issues that arise with SME and ARCS – what should the arity of a predicate be, and should single-place predicates (i.e., features or attributes) match? SME and ARCS represent attributes (i.e., features) as single-place predicates,[12] so one does not have to choose between representing something as an attribute or a predicate. However, the number of arguments of a predicate must be chosen, and the choice does have consequences. An attribute or feature like "small" can be represented as a one- or two-place predicate, e.g., as `is-small(mouse)` or `is-smaller-than(mouse, elephant)`. The choice is important for two reasons: neither SME nor ARCS allows matches between predicates with different numbers of arguments, and SME (in analogy mode) does not allow matches between single-place predicates, unless they are part of some larger structure. This is a general problem because a multi-place predicate can always be turned into a single-place predicate by moving arguments into the predicate name, e.g., `fed-peanuts-to-the-monkeys-on-tuesday(Jill)`.

## 6.4 Inadequacy of vector dot-product models of similarity

Tversky [1977] pointed out the now-well-known reasons why human similarity judgements cannot be modeled as a simple vector dot-product. The problems arise because there cases in which people's similarity judgements violate each of three axioms of metric spaces (with $d(\cdot,\cdot)$ as the distance function): minimality, $d(x,x) = 0$; symmetry, $d(x,y) = d(y,x)$; and the triangle inequality, $d(x,z) \leq d(x,y) + d(y,z)$.

However, these problems do not preclude the use of vector dot-products at some level in a model of similarity. Much of the problematic data could be explained by a vector dot-product model which allowed vectors to be transformed before computing the dot-product. The transform would depend on the context of the comparison and possibly on the items to be compared. It could be something as simple as an adjustment of the weighting of various components in the vector.

## 6.5 Scaling to have more episodes in memory

In the experiments reported here, the clean-up memory held only 12 episodes. However, many more episodes could be added without any decrease in accuracy of retrieval, provided the new episodes were not very similar to the existing ones. Retrieval or rating errors are most likely to occur with similar episodes; observed dot-products are distributed normally around the expected value, thus similarity errors are extremely unlikely with non-similar episodes because the tails of a normal distribution drop off so quickly. Thus, as long as any one episode is similar to only a small number of others, the system with $n = 2048$ or $n = 4096$ will be able to store many thousands of episodes.

## 6.6 Chunking, scaling, and networks of HRRs

One problem with the HRR dot-product as a measure of similarity is that it does not distinguish between a poorly matching pair of components and a well-matching but low-weighted pair of components. A possible solution to this could be to build a spreading activation network of chunks. Each episode would be represented by a collection of chunks, and each object and relation, and possibly even features, would be a separate chunk. For example, the chunks for "small-circle above large-square" could be as follows:

[12] Though ARCS's similarity matrix is an alternative location for representing attributes.

$$\mathbf{small}, \mathbf{circle}, \mathbf{smCircle}, \mathbf{large}, \mathbf{square}, \mathbf{lgSquare}, \mathbf{vertical},$$
$$\langle \mathbf{vertical} + \mathbf{above} \circledast \mathbf{smCircle} + \mathbf{below} \circledast \mathbf{lgSquare} \rangle, \text{ and}$$
$$\langle \langle \mathbf{smCircle} + \mathbf{lgSquare} \rangle + \langle \mathbf{vertical} + \mathbf{above} \circledast \mathbf{smCircle} + \mathbf{below} \circledast \mathbf{lgSquare} \rangle \rangle.$$

To build a network which matched this structure against another, we would need nodes for these chunks and those in the other structure, and we would build links between nodes whose strength was proportional to the dot-product similarity of the corresponding chunks. Some form of competition would be necessary to prevent all nodes going to maximum activation. The initial activation of chunks could reflect their a priori importance or salience. The network would be run by allowing it to settle to a stable state. The final state of the network would show which components of the structure were similar, and the degree-of-match between components could be easily extracted.

Another problem will probably occur when the HRR dot-product is used to estimate the similarity of episodes with many propositions. As recognized in SME and ARCS, a good analogy can occur when just one coherent part of one structure matches a part of the other. Thus, for larger episodes, a linear sum of matches of structural features is probably not a very appropriate measure of similarity. The dot-product measure does not distinguish between a large number of medium similarity matches and a small number of high similarity matches. Furthermore, while there is some sensitivity to the matches being localized in a part of the structure (due to the hierarchical binding chains), this effect is probably not very strong. Chunking, and using a spreading activation network of chunks is also a possible solution for this scaling problem. Large episodes would be stored as a number of chunks in long-term memory, and the chunks belong to the same episode would be quite similar. The probe episode would also consist of several chunks. The search for analogies to the probe in long-term memory would now consist of several steps. In the first step chunks that were similar to those of the probe would be retrieved. Next, chunks similar to retrieved chunks would be retrieved, in order to pull in entire (or large parts of) episodes from long-term memory. Next, a spreading activation network would be constructed as described above, and activation allowed to settle. The episode with the largest amount of activation would be considered the most similar. This process would still take structural similarity into account because chunks would not be similar unless the structures they came from were similar. It would also allow the weighting of various aspects (i.e., chunks) of each structure to change during and in response to the matching process. This would go a long way towards creating flexible connectionist model of similarity assessment like that speculated about by Gentner and Markman [1993].

## 6.7 Comparison to RAAMs and tensor-product models

Pollack's [1990] RAAMs are a type of backpropagation network which can learn to encode recursive structures in a fixed-width vector. Conceivably, one could get estimates of structural similarity by comparing the reduced representations discovered by RAAMs. Pollack shows some cluster diagrams which seem to have similar structures grouped together. However, the strength of this effect is unclear, and I am not aware of any systematic studies of this effect. Extrapolating from other things known about RAAMs, we could expect them to have two disadvantages relative to HRRs as a way of getting estimates of structural similarity:

1. RAAMs require much learning time and the generalization appears weak – there is no guarantee that a new structure can be represented at all (without extensive training). For example, if a RAAM were trained on episodes involving the predicates cause, bite, and flee, it might not even be able to represent episodes involving stroke or lick, let alone compare them to other episodes.

2. The non-linear nature of RAAMs makes it possible for similar structures to have quite different representations, even when the structures are similar in both surface and structural features.

On the other hand, the learning involved in RAAMs offers a potential advantage over HRRs – RAAMs could learn to devote more representational resources to commonly encountered structures and thus achieve better performance on them. The downside of this would be poorer performance on rarely encountered structures.

Smolensky's [1990] tensor-product representations are another method for encoding hierarchical structure in distributed representations. They have much in common with HRRs, and it would not be difficult to use them to replicate the results in this paper. The main difference between the two methods is that the dimensionality of the tensor-product representation increases exponentially with the depth of the structure, while the dimensionality of HRRs remains constant. This could lead to unreasonable resource requirements for structures with several levels of nesting. An advantage of tensor product representations is that they tend to be less noisy than HRRs in decoding, and would probably also give less noisy similarity estimates.

The results here would be difficult to replicate with Halford *et al*'s [to appear] representation for relations. Their representation for a relation is the tensor product of its arguments and name. For example, the relation `bite(Spot, Jane)` would be represented as $\mathbf{bite} \otimes \mathbf{spot} \otimes \mathbf{jane}$. One problem with this representation is that relations with just one dissimilar argument are dissimilar, where dissimilar means zero similarity. For example, $\mathbf{bite} \otimes \mathbf{spot} \otimes \mathbf{jane}$ is dissimilar to $\mathbf{bite} \otimes \mathbf{spot} \otimes \mathbf{fido}$ (assuming **jane** and **fido** are dissimilar). Another problem is that it is not clear how to represent episodes with multiple or nested propositions. Halford *et al* do mention chunking, but do not give any details how it might be done. On the other hand, the computations Halford *et al* perform are easy to do with a role-filler binding representation (using either tensor products or convolution), provided that relations are stored in a content-addressable item memory i.e., a clean-up memory.

## 6.8 Is it necessary to use circular convolution?

It is not necessary to use circular convolution as the associative operator to achieve the results in this paper. Circular convolution can be viewed as a two-argument hashing operator which happens to have a number of convenient properties:

1. Bilinear: $(\alpha\mathbf{a} + \beta\mathbf{b}) \circledast \mathbf{c} = \alpha\mathbf{a} \circledast \mathbf{c} + \beta\mathbf{b} \circledast \mathbf{c}$,[13] for all vectors $\mathbf{a}$, $\mathbf{b}$, and $\mathbf{c}$, and scalars $\alpha$ and $\beta$. This makes convolution similarity preserving: if $\mathbf{a}$ is similar to $\mathbf{a}'$ (i.e., $\mathbf{a}' = \mathbf{a} + \epsilon\mathbf{b}$ where $\epsilon$ is small) then $\mathbf{a} \circledast \mathbf{c}$ will be similar to $\mathbf{a}' \circledast \mathbf{b}$.

2. Invertible: if vectors elements are independently distributed as $\mathrm{N}(0, 1/n)$ there is a simple stable approximate inverse of circular convolution.[14] This makes it possible to extract components from HRRs.

3. Randomizing: the expected similarity of $\mathbf{a} \circledast \mathbf{b}$ to $\mathbf{a}$ or $\mathbf{b}$ is zero (provided that the expected similarity $\mathbf{a}$ and $\mathbf{b}$ with the identity (delta) vector is zero). This helps to avoid unwanted correlations.

4. Maps to same space: the circular convolution of two $n$-dimensional vectors is an $n$-dimensional vector. This allows easy representation of recursive structures.

5. Efficiently computable: the circular convolution of $n$-dimensional vectors can be computed in $O(n \log n)$ time using Fast Fourier transforms.

Convolution is also commutative and associative, but these properties are not essential.

[13] Convolution is commutative, so $\mathbf{c} \circledast (\alpha\mathbf{a} + \beta\mathbf{b}) = \alpha\mathbf{c} \circledast \mathbf{a} + \beta\mathbf{c} \circledast \mathbf{a}$ is also true.

[14] An exact inverse also exists, but it is less stable unless more constraints are placed on the vectors [Plate 1995].

Any associative operator which shares most of these properties will suffice. Other operators which could be used are matrix multiplication, permuted convolutions [Plate 1995], or disordered compressions of outer products [Plate 1994].

### 6.9 Relationship to representations in the brain

I do not with to propose that the the human brain uses HRRs as they are presented here. However, it could be a useful, and testable, hypothesis that the brain uses some type of representation akin to HRRs, in which bindings are represented by static activations vectors, and are constructed by a recursively-applicable associative operator similar to circular convolution.

Some type of compression of an outer product is an obvious candidate for the associative operator, but it seems unlikely that the highly regular compression defined by circular convolution would be found in neural tissue. However, it turns out that the regularity of the compression defined by convolution is not necessary for its function as an associative operator. In Plate [1994] I describe how a disordered compression of an outer product (i.e., sums of randomly selected elements of the outer product) can function as an associative operator. A disordered compression of the outer product can be implemented by randomly connected *sigma-pi* units, which were suggested by Feldman and Ballard [1982] as a of model interactions in the dendritic tree. This provides an initially plausible encoding operator for binding, and does not require any learning. Finding the corresponding decoding operator does require learning, but the function to be learned is simple (it is a linear map), and an unbounded amount of training data can be generated from random patterns.

### 6.10 Faster comparison with items in long-term memory

Computing the dot-product of the probe with each item in long-term memory can be expensive if there are a large number of items in long-term memory, especially on a serial machine. Much of this computation can be avoided by computing a fast estimate of the dot-product and then only computing the floating-point dot-product for the most promising candidates. The fast-estimate is the dot-product of thresholded vectors. Each item in long-term memory must have a thresholded versions stored with it. This is a vector of $n$ bits where the $n$th bit is one if the $n$th element of the item vector is positive, and zero otherwise. The dot-product of bit vectors can be computed many times faster, using full-word XOR operations, than the dot-product of floating point vectors. The bit-vector dot-product is very highly correlated with the floating-point dot-product – for $n = 2048$ the correlation coefficient was measured to be 0.997 (for 4096 pairs of vectors). This of course depends on HRR vector elements having a mean of zero, but this will be true if the HRRs are constructed following the descriptions in this paper. The speedup achieved will depend very much on the similarity of items in long-term memory, and on the details of implementation. Preliminary experiments, with many similar items, and with little effort put into optimizing the implementation, showed an improvement in computation time of at least 50% [Plate 1994].

## 7 Conclusion

While it would be overstating the importance of this work to describe it as a watershed for connectionist modeling, it does adds further evidence that complex structure and structural alignment is not the Waterloo for connectionist techniques.

The dot-product of HRRs provides a fast estimate of the degree of analogical similarity and is sensitive to various structural aspects of the match. It is not intended to be a model of complex

| Structural Similarity | Object Attribute Similarity YES | NO |
|---|---|---|
| YES | (LS) High | (AN) Low |
| NO | (SS) Med | (FA) Low |

(a) Ordinary-HRR dot-products.

| Structural Similarity | Object Attribute Similarity YES | NO |
|---|---|---|
| YES | (LS) High | (AN) ↕Med-High |
| NO | (SS) ↕Med-Low | (FA) Low |

(b) Contextualized-HRR dot-product.

Table 12:

or creative analogy making, but it could be a useful first stage in a model of analogical reminding. The HRR dot-product is only an estimate of analogical similarity for two reasons: its actual value is noisy, although the noise decreases with increasing vector dimension, and its expected value is an imperfect measure of analogical similarity.

The dot-product of ordinary HRRs is sensitive to some aspects of analogical similarity. It improves on the existing fast similarity matcher in MAC/FAC in that it discriminates the first column of Table 5(a) – it ranks literally similar (LS) episodes higher than superficially similar (SS) episodes. However, it is insensitive to object-level isomorphism when corresponding objects are not similar. Consequently, it ranks both analogies (AN) and false analogies (FA) lower than superficially similar (SS) episodes. The approximate values of the estimates it gives are shown in Table 12(a) – compare these with Table 5(a).

The dot-product of contextualized HRRs is sensitive to more aspects of structural similarity. In particular, it is sensitive to object-level isomorphism even when corresponding objects are not similar. It ranks the given examples in the same order as would SME or ARCS. The approximate values of its estimates are shown in Table 12(b) – these are the same as the SME and ARCS scores in Table 5(b).

HRR dot-products are flexible – the salience of various aspects of similarity can be adjusted by changing the weighting of various components in the probe. This is true for both ordinary and contextualized HRRs.

HRRs scale well with the number of episodes in long-term memory. Although the experiments here only had 12 episodes in long-term memory, many thousands more can be added without increasing the vector dimension, provided that no single episode is very similar to many others.

This work does to some extent contradict Gentner and Markman's [1993] implicit claim that performing structural alignment is a prerequisite to judging analogical similarity. No explicit structural alignment is performed when computing the HRR dot-product similarity estimate, yet the estimate is sensitive to the degree of structural alignment.

The HRR dot-product is not without its drawbacks. First, it is an imperfect measure of similarity, and there are examples where the expected values of HRR dot-products will result in rankings different from SME or ARCS. Second, high-dimensional vectors are required to reduce the noise in estimates to acceptable levels. Computing the dot-products of high-dimensional vectors can still require a significant amount of computation, though this computation can be reduced by filtering according to the bitwise dot-products (Section 6.10). Third, I do not expect the scaling with the number of propositions in an episode to be particularly good – the sum of structural-feature matches becomes a less appropriate measure of similarity as the episodes get larger. A possible solution to this problem is to construct a spreading activation network of HRRs in which each episode is represented as a number of chunks.

# References


Borsellino, A. and Poggio, T. 1973. Convolution and correlation algebras. *Kybernetik*, 13:113–122.

Falkenhainer, B., Forbus, K. D., and Gentner, D. 1989. The Structure-Mapping Engine: Algorithm and examples. *Artificial Intelligence*, 41:1–63.

Feldman, J. A. and Ballard, D. H. 1982. Connectionist models and their properties. *Cognitive Science*, 6:205–254.

Fodor, J. A. and Pylyshyn, Z. W. 1988. Connectionism and cognitive architecture: A critical analysis. *Cognition*, 28:3–71.

Gentner, D. and Forbus, K. D. 1991. MAC/FAC: A model of similarity-based retrieval. In *Proceedings of the Thirteenth Annual Cognitive Science Society Conference*, pages 504–509, Hillsdale, NJ. Erlbaum.

Gentner, D. and Markman, A. B. 1993. Analogy – Watershed or Waterloo? Structural alignment and the development of connectionist models of analogy. In Giles, C. L., Hanson, S. J., and Cowan, J. D., editors, *Advances in Neural Information Processing Systems 5 (NIPS*92)*, pages 855–862, San Mateo, CA. Morgan Kaufmann.

Gentner, D., Rattermann, M. J., and Forbus, K. D. 1993. The roles of similarity in transfer: Separating retrievability from inferential soundness. *Cognitive Psychology*, 25:431–467.

Goldstone, R. L., Medin, D. L., and Gentner, D. 1991. Relational similarity and the nonindependence of features in similarity judgements. *Cognitive Psychology*, 23:222–262.

Halford, G. S., Wilson, W. H., Guo, J., Gayler, R. W., Wiles, J., and Stewart, J. E. M. to appear. Connectionist implications for processing capacity limitations in analogies. In Holyoak, K. J. and Barnden, J., editors, *Analogical Connections*, volume 2 of *Advances in Connectionist and Neural Computation Theory*. Ablex, Norwood, NJ.

Hinton, G. E. 1990. Mapping part-whole heirarchies into connectionist networks. *Artificial Intelligence*, 46(1-2):47–76.

Hinton, G. E., McClelland, J. L., and Rumelhart, D. E. 1986. Distributed representations. In D. E. Rumelhart, J. L. M. and the PDP research group, editors, *Parallel distributed processing: Explorations in the microstructure of cognition.*, volume I, pages 77–109. Cambridge, MA: MIT Press.

Liepa, P. 1977. Models of content addressable distributed associative memory. Unpublished manuscript.

Markman, A. B., Gentner, D., and Wisniewski, E. J. 1993. Comparison and cognition: Implications of structure-sensitive processing for connectionist models. Unpublished manuscript.

Metcalfe Eich, J. 1982. A composite holographic associative recall model. *Psychological Review*, 89:627–661.

Murdock, B. B. 1982. A theory for the storage and retrieval of item and associative information. *Psychological Review*, 89(6):316–338.

Murdock, B. B. 1987. Serial-order effects in a distributed-memory model. In Gorfein, D. S. and Hoffman, R. R., editors, *MEMORY AND LEARNING: The Ebbinghaus Centennial Conference*, pages 277–310. Lawrence Erlbaum Associates.

Paek, E. G. and Psaltis, D. 1987. Optical associative memory using Fourier transform holograms. *Optical Engineering*, 26(5):428–433.

Plate, T. A. 1991. Holographic Reduced Representations: Convolution algebra for compositional distributed representations. In Mylopoulos, J. and Reiter, R., editors, *Proceedings of the 12th International Joint Conference on Artificial Intelligence*, pages 30–35, San Mateo, CA. Morgan Kaufmann.

Plate, T. A. 1994. *Distributed Representations and Nested Compositional Structure*. PhD thesis, Department of Computer Science, University of Toronto.

Plate, T. A. 1995. Holographic reduced representations. *IEEE Transactions on Neural Networks*, 6(3):623–641.

Pollack, J. B. 1990. Recursive distributed representations. *Artificial Intelligence*, 46(1-2):77–105.

Ross, B. 1989. Distinguishing types of superficial similarities: Different effects on the access and use of earlier problems. *Journal of Experimental Psychology Learning Memory Cognition*, 15:456–468.

Schönemann, P. H. 1987. Some algebraic relations between involutions, convolutions, and correlations, with applications to holographic memories. *Biological Cybernetics*, 56:367–374.

Seifert, C. M. and Hammond, K. J. 1989. Why there's no analogical transfer. In *Proceedings: Case-Based Reasoning Workshop*, pages 148–152. Morgan Kaufmann.

Smolensky, P. 1986. Neural and conceptual interpretation of PDP models. In D. E. Rumelhart, J. L. M. and the PDP research group, editors, *Parallel distributed processing: Explorations in the microstructure of cognition*, volume 2, pages 390–431. The MIT Press, Cambridge, MA.

Smolensky, P. 1990. Tensor product variable binding and the representation of symbolic structures in connectionist systems. *Artificial Intelligence*, 46(1-2):159–216.

Thagard, P., Holyoak, K. J., Nelson, G., and Gochfeld, D. 1990. Analog Retrieval by Constraint Satisfaction. *Artificial Intelligence*, 46:259–310.

Touretzky, D. S. 1990. Boltzcons: Dynamic symbol structures in a connectionist network. *Artificial Intelligence*, 42(1-2):5–46.

Tversky, A. 1977. Features of similarity. *Psychological Review*, 84(4):327–352.

Wharton, C. M., Holyoak, K. J., Downing, P. E., Lange, T. E., Wickens, T. D., and Melz, E. R. to appear. Below the surface: Analogical similarity and retrieval competition in reminding. *Cognitive Psychology*.

Willshaw, D. 1981. Holography, associative memory, and inductive generalization. In Hinton, G. E. and Anderson, J. A., editors, *Parallel models of associative memory*. Erlbaum, Hillsdale, NJ.

Willshaw, D. J., Buneman, O. P., and Longuet-Higgins, H. C. 1969. Non-holographic associative memory. *Nature*, 222:960–962.