Structured Operations with Distributed Vector Representations

Tony A. Plate
School of Mathematical and Computing Sciences, Victoria University of Wellington
Wellington, New Zealand
Email: tap@mcs.vuw.ac.nz

Abstract

Holographic Reduced Representations (HRRs) are a method for encoding nested relational structures in fixed width vector representations. HRRs encode relational structures as vector representations in such a way that the superficial similarity of the vectors reflects both superficial and structural similarity of the relational structures. HRRs support a number of operations that could be very useful in models of analogy processing: fast estimation of superficial and structural similarity via a vector dot-product; chunking of vector representations; and finding corresponding objects in two structures.

1 Introduction

Vector representations are popular for memory models for a variety of theoretical and practical reasons. They are simple and support fast parallel processing such comparison via dot-products. They are also neurologically plausible, in that they can be stored and processed in networks of simple neuron-like processing elements, such as associative vector memories. However, their use in models of analogy processing has been limited by the widespread supposition that it is difficult or impossible to encode compositional structure in vector representations (Fodor and Pylyshyn 1988, Ratcliff and McKoon 1989, Thagard, Holyoak, Nelson and Gochfeld, 1990, Gentner and Markman 1993,Forbus, Gentner and Law, 1994, Wharton *et al* 1994).

This supposition is false. Structure can be represented in vectors in a number of ways, e.g., Smolensky's (1990) tensor products, Pollack's (1990) RAAMs, Kanerva's (1996) binary spattercodes, and Plate's (1995) HRRs. This paper describes HRRs and makes a number of claims for their usefulness in models of analogy retrieval and processing:

- HRRs provide an adequate vector-based representation of structure (in contrast to feature-vector approaches, which must be complemented with a conventional symbolic representation for structure).
- Estimates of similarity that reflect both superficial and structural similarity can be computed quickly via vector dot-products. This technique shows similar abilities and limitations with respect to detecting similarities as are observed in people's ability to retrieve items from long term memory.
- Corresponding objects in two analogical structures can be found via fast but approximate vector-based techniques.
- HRRs provide an elegant implementation of chunking and "pointers" for complex, structured items stored as vectors in a content addressable memory.

2 Analogy processing in people

Analog retrieval and mapping have received a significant amount of attention in the psychological literature. Much attention has been devoted to teasing apart the differing effects of superficial and structural similarity in retrieval and mapping.

For illustrations, the following series of episodes are used in this paper. Together the episodes involve dogs (Fido, Spot and Rover), people (Jane, John and Fred), a cat (Felix) and a mouse (Mort). Members of one species are assumed to be similar to each other but not to members of other species. The "probe" episode, to which the others are compared, is "*Spot bit Jane, causing Jane to flee from Spot*". There are five other episodes, which have different combinations of types of similarity to the probe (all share predicates with the probe):

LS (Literal Similarity) "*Fido bit John, causing John to flee from Fido.*" (Has both structural and superficial similarity.)

SF (Surface features) "*John fled from Fido, causing Fido to bite John.*" (Has superficial but not structural similarity.)

CM (Cross-mapped analogy) "*Fred bit Rover, causing Rover to flee from Fred.*" (Has both structural and superficial similarity, but types of corresponding objects are switched.)

AN (Analogy) *"Mort bit Felix, causing Felix to flee from Mort."* (Has structural but not superficial similarity).

FOR (First-order-relations only) *"Mort fled from Felix, causing Felix to bite Mort."* (Has neither structural nor superficial similarity, other than shared predicates.)

It is generally accepted that in adults, structural similarity plays a large role in analogical mapping and conscious similarity judgements. The role of structural similarity in retrieval is less clear: some researchers argue that structural similarity usually has little effect on retrieval (Gentner, Rattermann, and Forbus 1993) while others argue that under some circumstances, structural similarity can influence retrieval (Wharton *et al* 1994). Others suggest that structural similarity matters only when the entities involved in the situations share superficial features (Ross 1989). Overall, the general pattern for retrievability of items from long term memory seems to be LS > CM ≥ SF > AN ≥ FOR.

Existing computational models of human performance on analog retrieval tasks such as ARCS (Thagard *et al* 1990), and MAC/FAC (Forbus, Gentner and Law, 1994) have explained the human retrieval data by invoking two processes. The first is a simple one based on superficial similarities. This explains much of the human performance, but cannot account for effects of structural similarity. Thus, these models require a second process that takes structural similarity into account, which involves additional complex computation. In this paper I will argue that HRRs can provide a single-stage model based on vector-matching that explains the pattern of retrieval ability observed in people.

3 Vector representations and operations

The two vector operations commonly used with vector representations are superposition (i.e., addition) and similarity (i.e., dot-product or cosine). These two vector operations, and other scalar-vector operations such as scaling and normalization, are sufficient for interesting and useful memory models. With the addition of the circular convolution operation for binding, one can encode associations in vector patterns which and thus encode structure.

3.1 Local & distributed representations

Vector representations come in two flavors: local and distributed. In some respects, localist and distributed representations are equivalent. They can be indistinguishable when features are numerous and fine-grained. Also, localist representations can be mapped to distributed ones by a simple linear map, and back by a thresholded linear map. However, a crucial difference is that the total number of possible features is limited to the vector dimensionality in localist representations, but is exponential in vector dimensionality in distributed representations. This gives distributed representations the capacity to represent combinatorial features (such as Wharton's *et al* (1994) sour-grapes feature "thing that is desired but can't be obtained and hence is denigrated") in a moderate sized vector.

What is needed is a systematic way of generating and decoding the patterns which represent combinatorial features. This is the role of the binding operation. As a binding operation, circular convolution provides a fast, systematic, and reversible way of constructing new patterns to represent combinatorial features.

3.2 Circular convolution

Circular convolution maps two real-valued n-dimensional vectors onto one. If **x** and **y** are n-dimensional vectors (subscripted 0 to n-1), then the elements of $\mathbf{z} = \mathbf{x} \otimes \mathbf{y}$ are

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

where subscripts are taken modulo-n. and ⊗ denotes circular convolution. Circular convolution can be viewed as a compression of the outer (or tensor) product of the two vectors, as shown in Figure 1. 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 **x** are the sums of the outer product elements along the wrapped diagonal lines.

Circular convolution can be regarded as a multiplication operator for vectors and has many algebraic properties in common with scalar and matrix multiplication. It is commutative ($\mathbf{x}\otimes\mathbf{y}=\mathbf{y}\otimes\mathbf{x}$), associative ($\mathbf{x}\otimes(\mathbf{y}\otimes\mathbf{z})=(\mathbf{x}\otimes\mathbf{y})\otimes\mathbf{z}$), and bilinear ($\mathbf{x}\otimes(\alpha\mathbf{y}+\beta\mathbf{z})=\alpha\mathbf{x}\otimes\mathbf{z}+\beta\mathbf{x}\otimes\mathbf{z}$). There is an identity vector **I** ($\mathbf{I}\otimes\mathbf{x}=\mathbf{x}$) and a zero vector $\bar{0}$ ($\bar{0}\otimes\mathbf{x}=\bar{0}$). Inverses $\mathbf{x}^{-1}$ exist for most vectors ($\mathbf{x}^{-1}\otimes\mathbf{x}=\mathbf{I}$).

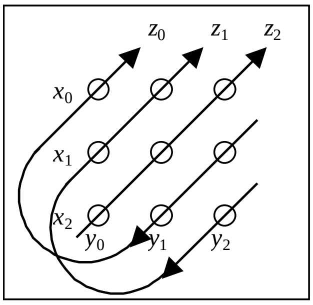

Figure 1

An association between two items $\mathbf{x}$ and $\mathbf{y}$ can be represented by the convolution of the two items: $\mathbf{x}\otimes\mathbf{y}$. The inverse vector of $\mathbf{x}$ can be used to reconstruct $\mathbf{y}$ from $\mathbf{x}\otimes\mathbf{y}$: $\mathbf{x}^{-1}\otimes(\mathbf{x}\otimes\mathbf{y})=\mathbf{y}$. However, except under certain restrictive conditions, the inverse is numerically unstable and is not always the best choice for decoding. For vectors which have randomly chosen elements independently distributed as N(0,1/*n*) (the normal distribution with mean 0 and variance 1/*n*) there is an approximate inverse with attractive properties. The approximate inverse of $\mathbf{x}$ is denoted by $\mathbf{x}^{\mathrm{T}}$. It is a simple rearrangement of the elements of $\mathbf{x}$: $x^{\mathrm{T}}_i = x_{-i}$, where subscripts are modulo-*n*. The approximate inverse is simple to compute and is numerically stable. Reconstruction using the approximation inverse is noisy. The convolution product $\mathbf{x}^{\mathrm{T}}\otimes\mathbf{x}\otimes\mathbf{y}$ can be written as $\mathbf{y}+\eta$, where the η can be considered as zero-mean noise whose magnitude (variance) decreases with increasing vector dimension.

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 $\mathbf{z}=\mathbf{x}\otimes\mathbf{y}+\mathbf{v}\otimes\mathbf{w}$. To find what is associated with $\mathbf{x}$ we convolve $\mathbf{z}$ with $\mathbf{x}^{\mathrm{T}}$. The result can be expressed as $\mathbf{x}^{\mathrm{T}}\otimes\mathbf{x}\otimes\mathbf{y}+\mathbf{x}^{\mathrm{T}}\otimes\mathbf{v}\otimes\mathbf{w}$. The first term 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 $\mathbf{y}$.

3.3 Similarity preservation and randomization

Convolution preserves both similarity and lack of similarity in a multiplicative fashion: the similarity of two role-filler binding patterns is approximately equal to the product of the similarities of the respective role and filler patterns (provided that the role patterns are not similar to the filler patterns.) Thus, if two bindings have the same role, their similarity will be equal to that of the fillers. Conversely, if two roles have no similarity, bindings involving them will have similarity regardless of the fillers. Furthermore, convolution is randomizing in that role-filler binding patterns are not similar to either the role or filler patterns.

3.4 HRRs for relational structure

Consider representing a nested proposition such as “Spot bit Jane, causing Jane to flee from Spot” in a vector pattern. We would like this pattern to faithfully record structure and also to be suitable for detecting at least superficial similarity by computing dot-products.

The structure of a proposition can be represented by superimposing patterns representing the predicate name and the role-filler bindings. This provides a structural skeleton that faithfully records structure.

The skeleton HRR for the proposition “Spot bit Jane” is constructed as follows:

$$\mathbf{K_{P\text{-}bite}}=\mathbf{bite}+\mathbf{bite_{agt}}\otimes\mathbf{spot}+\mathbf{bite_{obj}}\otimes\mathbf{jane}$$

The pattern **bite** represents the predicate label, $\mathbf{bite}_{\mathrm{agt}}$ and $\mathbf{bite}_{\mathrm{obj}}$ its roles, and **spot** and **jane** the entities “Spot” and “Jane”. If we have the pattern $\mathbf{K_{P\text{-}bite}}$ and know the role patterns, then we can reconstruct the filler patterns by convolving $\mathbf{K_{P\text{-}bite}}$ with the approximate inverses of the role patterns. For example, $\mathbf{bite}_{\mathrm{agt}}{}^{\mathrm{T}}\otimes\mathbf{K_{P\text{-}bite}}$ gives a noisy version of **spot** which, if necessary, can be cleaned up using an auto-associative item memory. The pattern **bite** is made a component of $\mathbf{K_{P\text{-}bite}}$ in order to identify it as a *bite* proposition and thus allow a system to deduce that the appropriate role patterns for decoding are $\mathbf{bite}_{\mathrm{agt}}$ and $\mathbf{bite}_{\mathrm{obj}}$.

The skeleton HRR pattern for the proposition “Spot bit Jane” is an *n*-dimensional pattern just like the patterns **spot**, **bite**, etc. Thus, it is easily used as a filler in a higher-order proposition. For example, the skeleton HRR $\mathbf{K_P}$ representing “Spot bit Jane, which caused Jane to flee from Spot” is constructed as follows:

$$\mathbf{K_{P\text{-}flee}}=\mathbf{flee}+\mathbf{flee_{agt}}\otimes\mathbf{jane}+\mathbf{flee_{from}}\otimes\mathbf{spot}$$

$$\mathbf{K_{P}}=\mathbf{cause}+\mathbf{cause}_{\mathrm{antc}}\otimes\mathbf{K_{P\text{-}bite}}+\mathbf{cause}_{\mathrm{cnsq}}\otimes\mathbf{K_{P\text{-}flee}}$$

The other goal for a vector representation was that patterns should reflect superficial similarity, i.e., two patterns should be similar if the structures they represent merely involve similar fillers or predicates. The presence of predicate labels in HRRs ensures that patterns for the same predicate are similar. However, skeleton HRRs do not behave as desired with respect to the presence of similar fillers: the randomizing properties of convolution mean that $\mathbf{role_1}\otimes\mathbf{filler_1}$ is only similar to $\mathbf{role_2}\otimes\mathbf{filler_2}$ to the extent that $\mathbf{role_1}$ is similar to $\mathbf{role_2}$ *and* $\mathbf{filler_1}$ is similar to $\mathbf{filler_2}$. HRRs are easily made to reflect superficial similarity by superimposing the filler patterns together with the structural skeleton HRR. Thus, the fleshed-out HRR for "Spot bit Jane" is as follows:

$$\begin{aligned}\mathbf{P_{bite}} = \; & \mathbf{bite} + \mathbf{spot} + \mathbf{jane} \\ & + \mathbf{bite}_{\mathrm{agt}}\otimes\mathbf{spot} + \mathbf{bite}_{\mathrm{obj}}\otimes\mathbf{jane}\end{aligned}$$

Adding in the fillers makes decoding more noisy, but does not prevent successful decoding. For higher level propositions, the same idea of adding in fillers can be applied recursively. For example, the HRR for "Spot bit Jane, causing Jane to flee from Spot" is constructed as follows:

$$\begin{aligned}\mathbf{P_{flee}} = \; & \mathbf{flee} + \mathbf{spot} + \mathbf{jane} \\ & + \mathbf{flee}_{\mathrm{agt}}\otimes\mathbf{jane} + \mathbf{flee}_{\mathrm{from}}\otimes\mathbf{spot} \\ \mathbf{P} = \; & \mathbf{cause} + \mathbf{P_{bite}} + \mathbf{P_{flee}} \\ & + \mathbf{cause}_{\mathrm{antc}}\otimes\mathbf{P_{bite}} + \mathbf{cause}_{\mathrm{cnsq}}\otimes\mathbf{P_{flee}}\end{aligned}$$

HRRs constructed like this will be similar if they merely involve similar entities or predicates. Because of the similarity preserving properties of convolution, they will be even more similar if the entities are involved in similar roles.

3.5 The need for a "clean-up" memory

Convolution encodings are remarkably compact: a number of associations between *n*-dimensional patterns packed into one *n*-dimensional pattern. The price we pay for this compactness is noise in decoded vectors. Consequently, if we want a convolution-based associative-memory model to provide accurate reconstructions of decoded patterns, it must be equipped with an additional error-correcting auto-associative item memory. This can clean up the noisy patterns retrieved from the convolution encodings. This clean-up memory must store all the items that the system can produce. When given a noisy version of one of those items it must either output the closest item or indicate that the input is not close to any of the stored items. Note that only a few associations are stored as convolution encodings in a single pattern, whereas many patterns are stored in the clean-up memory.

3.6 Normalization

The final point to consider when constructing HRRs is maintaining the overall strength of patterns and the statistical distribution of their elements. The easiest way to do this is to normalize all patterns to have a Euclidean length of one. Here, the normalized version of the vector $\mathbf{x}$ is denoted by $\langle\mathbf{x}\rangle$ and is defined as follows:

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

4 Estimating Similarity

The six "dog bites human" episodes provide a simple demonstration that HRR scores can reflect similarity of structural arrangements, as well as similarity of surface features. It also demonstrates that a model based just on HRR similarity scores can neatly explain the pattern of human retrieval: LS > CM ≥ SF > AN ≥ FOR.

The HRRs for the probe ($\mathbf{P}$) and the literally similar episode ($\mathbf{E_{LS}}$) are constructed as follows:

$$\begin{aligned}\mathbf{P_{bite}} = \; & \langle \mathbf{bite} + \langle \mathbf{spot} + \mathbf{jane} \rangle \\ & + \mathbf{bite}_{\mathrm{agt}}\otimes\mathbf{spot} + \mathbf{bite}_{\mathrm{obj}}\otimes\mathbf{jane} \rangle \\ \mathbf{P_{flee}} = \; & \langle \mathbf{flee} + \langle \mathbf{spot} + \mathbf{jane} \rangle \\ & + \mathbf{flee}_{\mathrm{agt}}\otimes\mathbf{jane} + \mathbf{flee}_{\mathrm{from}}\otimes\mathbf{spot} \rangle \\ \mathbf{P} = \; & \langle \mathbf{cause} + \langle \mathbf{P_{bite}} + \mathbf{P_{flee}} \rangle \\ & + \mathbf{cause}_{\mathrm{antc}}\otimes\mathbf{P_{bite}} + \mathbf{cause}_{\mathrm{cnsq}}\otimes\mathbf{P_{flee}} \rangle \\ \mathbf{E_{LS\text{-}bite}} = \; & \langle \mathbf{bite} + \langle \mathbf{spot} + \mathbf{jane} \rangle \\ & + \mathbf{bite}_{\mathrm{agt}}\otimes\mathbf{spot} + \mathbf{bite}_{\mathrm{obj}}\otimes\mathbf{jane} \rangle \\ \mathbf{E_{LS\text{-}flee}} = \; & \langle \mathbf{flee} + \langle \mathbf{spot} + \mathbf{jane} \rangle \\ & + \mathbf{flee}_{\mathrm{agt}}\otimes\mathbf{jane} + \mathbf{flee}_{\mathrm{from}}\otimes\mathbf{spot} \rangle \\ \mathbf{E_{LS}} = \; & \langle \mathbf{cause} + \langle \mathbf{P_{bite}} + \mathbf{P_{flee}} \rangle \\ & + \mathbf{cause}_{\mathrm{antc}}\otimes\mathbf{P_{bite}} + \mathbf{cause}_{\mathrm{cnsq}}\otimes\mathbf{P_{flee}} \rangle\end{aligned}$$

The HRRs for the other episodes are built in an analogous fashion. The patterns for members of the same species (types) are designed to be similar. The complete set of base vectors and tokens used in this experiment is shown in Table 1. All base and identity (**id**) vectors were randomly chosen with elements independently distributed as N(0,1/*n*).

Average HRR similarity scores are shown in Table 2. These are from 100 runs with different random base and identity vectors, and a vector dimension of 2048. The directions of differences between average similarity scores were reliable – the standard deviation of the scores ranged between 0.016 and 0.026.

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		$\mathbf{spot} = \langle \mathbf{dog} + \mathbf{id}_{spot} \rangle$
		$\mathbf{fido} = \langle \mathbf{dog} + \mathbf{id}_{fido} \rangle$
$\mathbf{bite}_{agt}$	$\mathbf{bite}_{obj}$	$\mathbf{rover} = \langle \mathbf{dog} + \mathbf{id}_{rover} \rangle$
$\mathbf{flee}_{agt}$	$\mathbf{flee}_{obj}$	$\mathbf{felix} = \langle \mathbf{cat} + \mathbf{id}_{felix} \rangle$
$\mathbf{cause}_{antc}$	$\mathbf{cause}_{cnsq}$	$\mathbf{mort} = \langle \mathbf{mouse} + \mathbf{id}_{mort} \rangle$

Table 1

For comparison, MAC-style similarity scores are also shown. These are modeled after the MAC stage of Forbus *et al*'s (1994) MAC/FAC model. They are based on the dot product of normalized content vectors over the following features: *person, dog, mouse, cat, cause, bite,* and *flee.* For example, the content vector for the probe is $(1,1,0,0,1,1,1)/\sqrt{5}$.

The pair of episodes $\mathbf{E}_{LS}$ and $\mathbf{E}_{SF}$ each have the same surface commonalities (object features and predicate names) with the probe. The difference between them is that $\mathbf{E}_{LS}$ is structurally isomorphic to the probe, while $\mathbf{E}_{SF}$ is not. Because there is no structural information beyond predicates names encoded in content vectors, $\mathbf{E}_{LS}$ and $\mathbf{E}_{SF}$ have the same content-vector similarity to the probe. On the other hand, the HRR similarity scores indicates that $\mathbf{E}_{LS}$ is more similar to the probe than $\mathbf{E}_{SF}$.

When episodes do not share object attributes with the probe, HRR scores are low and do not always reflect structural match. Although in Table 2 the HRR score for $\mathbf{E}_{AN}$ is higher than for $\mathbf{E}_{FOR}$ (due to the "bite" and "flee" propositions filling the same roles in $\mathbf{E}_{AN}$ as in the probe), this difference is not reliable. It is possible to construct other FOR examples that have a higher score than AN examples (Plate, 1994).

$\mathbf{E}_{CM}$ is a cross-mapped analogy. It has the same structure and types of objects as the probe, but unlike $\mathbf{E}_{LS}$ and the probe, the similar objects do not map to each other (the dog maps to the person, and the person maps to the dog). Since HRR similarity scores are sensitive to having similar objects fill similar roles, $\mathbf{E}_{CM}$ has a lower HRR similarity to the probe than $\mathbf{E}_{LS}$. In contrast, the content-vector similarities of $\mathbf{E}_{CM}$ and $\mathbf{E}_{LS}$ to the probe are the same.

4.1 Why HRR dot-products reflect structural similarity

HRR dot-products reflect structural similarity because of the presence of components representing combinatorial features, such as $\mathbf{bite}_{agt} \otimes \mathbf{spot}$, $\mathbf{cause}_{antc} \otimes \mathbf{bite}$, and $\mathbf{cause}_{antc} \otimes \mathbf{bite}_{agt} \otimes \mathbf{spot}$.

All of these higher-order features derive from role-filler bindings. Consequently, the HRRs described here reflect differences in structural similarity when there are differences in whether similar objects fill similar roles. Hence the large difference between $\mathbf{E}_{CM}$ and $\mathbf{E}_{LS}$ in their HRR similarity scores with the probe. Although they have the same objects, and isomorphic structure, $\mathbf{E}_{CM}$ does not have similar objects filling the same roles as in the probe. Thus, $\mathbf{E}_{CM}$ has combinatorial features like $\mathbf{bite}_{agt} \otimes \mathbf{person}$, which are not at all similar to those like $\mathbf{bite}_{agt} \otimes \mathbf{dog}$.

This pattern of sensitivity to structural similarity, in which structural similarity is only detected when similar objects fill similar roles, is very similar to the pattern observed by Ross (1989) in experiments with people. Ross found that shared structure enhanced retrieval in the presence of similar objects, provided that corre-

		Commonalities with probe			Similarity scores	
P	Spot bit Jane, causing Jane to flee from Spot.					
	Episodes in long-term memory:	Object attributes	First-order relation names	Higher-order structure	HRR	MAC
$\mathbf{E}_{LS}$	Fido bit John, causing John to flee from Fido.	✓	✓	✓	0.71	1.0
$\mathbf{E}_{SF}$	John fled from Fido, causing Fido to bite John	✓	✓	×	0.47	1.0
$\mathbf{E}_{CM}$	Fred bit Rover, causing Rover to flee from Fred.	✓	✓	✓	0.47	1.0
$\mathbf{E}_{AN}$	Mort bit Felix, causing Felix to flee from Mort.	×	✓	✓	0.42	0.6
$\mathbf{E}_{FOR}$	Mort fled from Felix, causing Felix to bite Mort.	×	✓	×	0.30	0.6

Table 2

sponding objects were similar, and that cross-mapping inhibited retrieval.

5 Interpretations of an Analogy

Retrieval of analogies is only the first step in many analogy processing tasks. After retrieving a potentially analogous episode we may want to decode the structure in order to evaluate more accurately the degree of structural consistency, or to use the episode for analogical reasoning. The structure of a HRR could be decoded using the techniques described in Section 2, and then used in a symbolic processor like SME or in some other connectionist architecture. However, some apparently more symbolic tasks, like finding corresponding entities, and thus deriving an interpretation of an analogy, can be computed with vector operations directly on HRRs.

Consider the probe **P** "Spot bit Jane, causing Jane to flee from Spot", and $\mathbf{E}_{\mathbf{LS}}$ "Fido bit John, causing John to flee from Fido." The entity corresponding to Jane (which is John) can be found in two steps:

1. Extract the roles Jane fills in the probe with the operation:

$$\mathbf{jane\text{-}roles} = \langle \mathbf{P} \otimes \mathbf{jane}^{\mathrm{T}} \rangle$$

This pattern is a blend of various roles and other noise patterns. The following are the positive dot-products of the **jane-roles** pattern with other role patterns:

$$\mathbf{jane\text{-}roles} \cdot \mathbf{cause}_{\mathrm{antc}} = 0.20$$
$$\mathbf{jane\text{-}roles} \cdot \mathbf{cause}_{\mathrm{cnsq}} = 0.18$$
$$\mathbf{jane\text{-}roles} \cdot \mathbf{flee}_{\mathrm{agt}} = 0.13$$
$$\mathbf{jane\text{-}roles} \cdot \mathbf{bite}_{\mathrm{obj}} = 0.12$$

2. Use **jane-roles** to extract the fillers from $\mathbf{E}_{\mathbf{LS}}$ and compare with the entities in $\mathbf{E}_{\mathbf{LS}}$:

$$\langle \mathbf{E}_{\mathbf{LS}} \otimes \mathbf{jane\text{-}roles}^{\mathrm{T}} \rangle \cdot \mathbf{john} = 0.38$$
$$\langle \mathbf{E}_{\mathbf{LS}} \otimes \mathbf{jane\text{-}roles}^{\mathrm{T}} \rangle \cdot \mathbf{fido} = 0.05$$

The most similar pattern is **john**, which is in fact the entity in $\mathbf{E}_{\mathbf{LS}}$ corresponding to Jane.

$\langle \mathbf{E}_{\mathbf{LS}} \otimes \mathbf{jane\text{-}roles}^{\mathrm{T}} \rangle$	**john** **fido**	0.38 0.07	✓
$\langle \mathbf{E}_{\mathbf{CM}} \otimes \mathbf{jane\text{-}roles}^{\mathrm{T}} \rangle$	**fred** **rover**	0.25 0.17	×
$\langle \mathbf{E}_{\mathbf{AN}} \otimes \mathbf{jane\text{-}roles}^{\mathrm{T}} \rangle$	**felix** **mort**	0.16 0.09	✓
$\langle \mathbf{E}_{\mathbf{SF}} \otimes \mathbf{jane\text{-}roles}^{\mathrm{T}} \rangle$	**john** **fido**	0.23 0.07	?
$\langle \mathbf{E}_{\mathbf{FOR}} \otimes \mathbf{jane\text{-}roles}^{\mathrm{T}} \rangle$	**mort** **felix**	0.11 0.06	?

Table 3

Table 3 shows the extraction of the entities corresponding to Jane in the various episodes. Correct extractions are checkmarked, and cases where there is no clear corresponding object have a question mark.

The correct answer is obtained in $\mathbf{E}_{\mathbf{LS}}$, where corresponding objects are similar, and in $\mathbf{E}_{\mathbf{AN}}$, where there is no object similarity. This extraction process has a bias towards choosing similar entities as the corresponding ones, which leads to a reasonable answer for $\mathbf{E}_{\mathbf{SF}}$ and an incorrect answer $\mathbf{E}_{\mathbf{CM}}$. There are no correct answers for $\mathbf{E}_{\mathbf{SF}}$ and $\mathbf{E}_{\mathbf{FOR}}$, because there are no consistent mapping between **P** and those episodes. However, because of the bias for mapping similar items, Fred is strongly indicated to be the entity in $\mathbf{E}_{\mathbf{SF}}$ corresponding to Jane. The only wrong answer is given for the cross-mapped analogy $\mathbf{E}_{\mathbf{CM}}$ where again the more similar object is indicated to be the corresponding one. Again, the effect of cross-mapping is similar to that observed by Ross (1989) in people: cross-mapping causes less accurate mapping performance.

Closer examination of the extraction process reveals both the reason for this bias and several ways of eliminating it, if that should be desired. Consider patterns containing just two of the components from **P** and $\mathbf{E}_{\mathbf{CM}}$:

$$\mathbf{P}' = \mathbf{cause} + \mathbf{bite}_{\mathrm{obj}} \otimes \mathbf{jane}$$
$$\mathbf{E}'_{\mathbf{CM}} = \mathbf{cause} + \mathbf{bite}_{\mathrm{obj}} \otimes \mathbf{rover}$$

The roles of Jane in $\mathbf{P}'$ are computed as:

$$\begin{aligned} \mathbf{jane\text{-}roles}' &= \mathbf{P}' \otimes \mathbf{jane}^{\mathrm{T}} \\ &\approx \mathbf{cause} \otimes \mathbf{jane}^{\mathrm{T}} + \mathbf{bite}_{\mathrm{obj}} \end{aligned}$$

The role pattern $\mathbf{bite}_{\mathrm{obj}}$ (and other role patterns like $\mathbf{flee}_{\mathrm{agt}}$ and $\mathbf{cause}_{\mathrm{antc}}$ in the full version of $\mathbf{P} \otimes \mathbf{jane}^{\mathrm{T}}$) are what are wanted here. The other patterns like $\mathbf{cause} \otimes \mathbf{jane}^{\mathrm{T}}$, which are not roles at all, are the source of the same-type bias in finding the corresponding object. When $\mathbf{jane\text{-}roles}'^{\mathrm{T}}$ is used to extract the fillers from $\mathbf{E}_{\mathbf{CM}}$, we get the following:

$$\begin{aligned} \mathbf{corresp} &= \mathbf{jane\text{-}roles}'^{\mathrm{T}} \otimes \mathbf{E}'_{\mathbf{CM}} \\ &\approx (\mathbf{cause} \otimes \mathbf{jane}^{\mathrm{T}} + \mathbf{bite}_{\mathrm{obj}})^{\mathrm{T}} \\ &\quad \otimes (\mathbf{cause} + \mathbf{bite}_{\mathrm{obj}} \otimes \mathbf{rover}) \\ &= \underline{\mathbf{jane}} + \mathbf{cause} \otimes \mathbf{jane}^{\mathrm{T}} \otimes \mathbf{bite}_{\mathrm{obj}} \otimes \mathbf{rover} \\ &\quad + \mathbf{bite}_{\mathrm{obj}}{}^{\mathrm{T}} \otimes \mathbf{cause} + \underline{\mathbf{rover}} \end{aligned}$$

This includes the pattern **rover** as desired, but also includes the pattern **jane** (from $(\mathbf{cause} \otimes \mathbf{jane}^{\mathrm{T}})^{\mathrm{T}} \otimes \mathbf{cause}$). Although $\mathbf{corresp}'$ only contains one term like this, there is a **jane** component in **corresp** for every pattern which is shared by **P** and $\mathbf{E}_{\mathbf{CM}}$. This adds up to a very strong component of **jane** in **corresp**. When **corresp** is compared to the fillers of $\mathbf{E}_{\mathbf{CM}}$, **cor-**

resp is more similar to **fred** than **rover**, due to the strong **jane** component in **corresp**.

One way of eliminating this similar-type bias is to perform a linear, multi-way, role-clean-up on **jane-roles**. This should pass all positive role components and suppress negative role and non-role components like $\mathbf{cause}\otimes\mathbf{jane}^{T}$. Thus, the clean version of **jane-roles** is as follows:

$$\begin{aligned}\mathbf{clean\text{-}jane\text{-}roles} = {} & \\ & 0.20\cdot\mathbf{cause}_{antc} + 0.18\cdot\mathbf{cause}_{cnsq} \\ + {} & 0.13\cdot\mathbf{flee}_{agt} + 0.12\cdot\mathbf{bite}_{obj}\end{aligned}$$

$\langle \mathbf{E}_{LS}\otimes\mathbf{cleaned\text{-}jane\text{-}roles}^{T}\rangle$	**john** **fido**	0.27 0.20	✓
$\langle \mathbf{E}_{CM}\otimes\mathbf{cleaned\text{-}jane\text{-}roles}^{T}\rangle$	**fred** **rover**	0.20 0.29	 ✓
$\langle \mathbf{E}_{AN}\otimes\mathbf{cleaned\text{-}jane\text{-}roles}^{T}\rangle$	**felix** **mort**	0.25 0.20	✓
$\langle \mathbf{E}_{SF}\otimes\mathbf{cleaned\text{-}jane\text{-}roles}^{T}\rangle$	**john** **fido**	0.25 0.17	?
$\langle \mathbf{E}_{FOR}\otimes\mathbf{cleaned\text{-}jane\text{-}roles}^{T}\rangle$	**mort** **felix**	0.26 0.19	?

Table 4.

The corresponding objects extracted using role clean-up are shown in Table 4. This slightly slower process gives correct answers for the episodes in which there is a consistent mapping.

The other way of avoiding the similar-type bias is to use a different binding operation, in which the algebraic properties of encoding and decoding do not result in terms like $(\mathbf{cause}\otimes\mathbf{jane}^{T})^{T}\otimes\mathbf{cause}$ equating to **jane**. Possible suitable alternative binding operations are discussed in Plate (1994).

There are two more limitations with these fast techniques for deriving interpretations. One is that each corresponding pair in a mapping is extracted independently. This matters when there is more than one consistent mapping. For example, if we have two possible consistent 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, but this will not be the case with the above techniques. To overcome this problem requires some other mechanism for checking that a mapping is one-to-one. The other problem is that these techniques fail when two different objects have the same set of roles – in such a case ambiguous results can be produced.

6 Chunking & memory organization

HRRs provide a natural method for chunking. In fact, a model based on HRRs must use chunking if it is to store structures of unlimited size. Chunking 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 **P** we first extract the cause antecedent pattern. This gives a noisy version of $\mathbf{P}_{\mathbf{bite}}$, which can be cleaned up by accessing item memory and retrieving the closest match. Now we have an accurate version of $\mathbf{P}_{\mathbf{bite}}$ from which we can extract the filler of the agent role.

To use chunks there must be a way of referring, or pointing to the chunks. In content-addressable memory in general, “pointers” to sub-chunks cannot be addresses, but must somehow hint at the contents of the sub-chunk. In HRRs, a decoded filler or sub-chunk, which is derived from a chunk by decoding with a role pattern, functions as an associative “pointer” to a pattern in item memory. These associative pointers are different from conventional pointers in that their form conveys information about their referent, information that is noisy but immediately available without the need to access memory. The advantage of having pointers that encode information about their referents 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 them.

6.1 Overall memory organization

In a system that uses HRRs there must be two levels of memory organization. One level encodes the structure in and among chunks. The other level stores large numbers of chunks (the large-scale clean-up memory).

Convolution encoding is most suited for encoding structure in and among small chunks in memory. Because of its memory capacity characteristics and noise in retrieval, convolution does not provide a suitable associative memory technique for the clean-up memory, which must store all the chunks. For this purpose we require some sort of large-scale error-correcting auto-associative memory. This large-scale memory should have the following properties:

auto-associative & error-correcting ability: when given a pattern, it should return accurately the closest one(s) stored, and

high capacity: the number of patterns which can be stored should be exponential in the size of the patterns.

There are several ways the clean-up memory could be implemented, e.g., Kanerva's (1988) sparse distributed memory, and Baum *et al*'s (1988) various content addressable schemes.

7 Discussion

This paper has described a scheme for encoding structure in vector representations based on circular convolution. Other approaches, such as Smolensky's (1990) tensor products, Pollack's (1990) RAAMs, Kanerva's (1996) binary spattercodes, have much in common – see Plate (1997) for a discussion – and could also be used in models of analogy processing.

The origin of patterns representing types such as 'dog', 'cat' and 'human' must be addressed at some stage. One possible automatic technique for learning such patterns is Latent Semantic Analysis (LSA), which learns high-dimensional vector patterns for words from large quantities of text (Landauer, Laham, and Foltz 1998). These patterns reflect human similarity judgements and could easily be used with HRRs.

The existence of a fast technique for computing good guesses at object correspondences suggests a new model for analogical mapping. Mapping could be done by "guessing" correspondences while stepping through the components of two structures and verifying that the proposed correspondences are consistent. This would require three mechanisms, one for traversing structures, another for guessing correspondences, and the last for storing correspondences and checking their consistency. All can be implemented with operations on vector representations. Such a model differs from ACME and SME in that it puts complexity at a different level. The top level involves simple sequential computation (traversing a structure and checking for mapping inconsistencies) rather than complex structural matching or construction of special networks, while the bottom level involves information-rich vector processing to measure similarities and estimate correspondences.

8 Conclusion

Holographic Reduced Representations provide a useful vector representation for analog retrieval and processing tasks. They provide chunking, which will be essential in vector-based model that stores large structures. They also support fast operations for computing similarity and object correspondences. These fast operations appear to have the right amount of power for modeling human abilities: their strengths and weaknesses follow a similar pattern to human performance on various analogy tasks. In particular, HRRs provide a simple, single-stage model of human performance on analog retrieval: HRR dot-products are sensitive to superficial similarity, and also to structural similarity in situations where corresponding roles have similar fillers, which is the same pattern of performance as demonstrated by human subjects on analog retrieval tasks.

9 References

Baum, E. B., J. Moody, and F. Wilczek (1988). Internal representations for associative memory. *Biological Cybernetics* 59, 217–228.

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

Forbus, K. D., D. Gentner, and K. Law (1994). MAC/FAC: A model of similarity-based retrieval. *Cognitive Science* 19, 141–205.

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

Gentner, D., M. J. Rattermann, and K. D. Forbus (1993). The roles of similarity in transfer: Separating retrievability from inferential soundness. *Cognitive Psychology* 25(4), 524–575.

Hinton, G. E., J. L. McClelland, and D. E. Rumelhart (1986). Distributed representations. In D. E. Rumelhart, J. L. McClelland, and the PDP research group (Eds.), *Parallel distributed processing: Explorations in the microstructure of cognition*, Volume 1, 77–109. MIT Press.

Kanerva, P. (1988). *Sparse Distributed Memory*. MIT Press.

Kanerva, P. (1996). Binary spatter-coding of ordered k-tuples. In C. von der Malsburg, W. von Seelen, J. Vorbruggen, and B. Sendhoff (Eds.), Artificial Neural Networks–ICANN Proceedings, Volume 1112 of *Lecture Notes in Computer Science, Berlin*, 869–873. Springer.

Landauer, T. K., D. Laham and P. W. Foltz (1998). Learning Human-like Knowledge with Singular Value Decomposition: A Progress Report. In *Neural Information Processing Systems* (NIPS*97). MIT Press.

Plate, T. A. (1994). Distributed Representations and Nested Compositional Structure. Ph.D. thesis, Dept. of Computer Science, University of Toronto. Available at http://www.mcs.vuw.ac.nz/~tap.

Plate, T. A. (1995). Holographic reduced representations. *IEEE Transactionls on Neural Neo*