

REASONING IN VECTOR SPACE: AN EXPLORATORY STUDY OF QUESTION ANSWERING

Moontae Lee*

Department of Computer Science
Cornell University University
Ithaca, NY 14850, USA
moontae@cs.cornell.edu

Xiaodong He, Wen-tau Yih, Jianfeng Gao & Li Deng

Microsoft Research
Redmond, WA 98052, USA
{xiaohe, scottyih, jfgao, deng}@microsoft.com

Paul Smolensky*

Department of Cognitive Science
Johns Hopkins University
Baltimore, MD 21218, USA
smolensky@jhu.edu

ABSTRACT

Question answering tasks have shown remarkable progress with distributed vector representation. In this paper, we investigate the recently proposed Facebook bAbI tasks which consist of twenty different categories of questions that require complex reasoning. Because the previous work on bAbI are all end-to-end models, errors could come from either an imperfect understanding of semantics or in certain steps of the reasoning. For clearer analysis, we propose two vector space models inspired by Tensor Product Representation (TPR) to perform knowledge encoding and logical reasoning based on common-sense inference. They together achieve near-perfect accuracy on all categories including positional reasoning and path finding that have proved difficult for most of the previous approaches. We hypothesize that the difficulties in these categories are due to the multi-relations in contrast to uni-relational characteristic of other categories. Our exploration sheds light on designing more sophisticated dataset and moving one step toward integrating transparent and interpretable formalism of TPR into existing learning paradigms.

1 INTRODUCTION

Ideal machine learning systems should be capable not only of learning rules automatically from training data, but also of transparently incorporating existing principles. While an end-to-end framework is suitable for learning without human intervention, existing human knowledge is often valuable in leveraging data toward better generalization to novel input. Question answering (QA) is one of the ultimate tasks in Natural Language Processing (NLP) on which synergy between the two capabilities could enable better understanding and reasoning.

Recently the Facebook bAbI tasks were introduced to evaluate complex reading comprehension via QA (Weston et al. (2015)); these have received considerable attention. Understanding natural questions, for example in WebQuestions tasks (Berant et al. (2013)), requires significant comprehension of the semantics, yet reasoning out the answers is then relatively simple (e.g., Bordes et al. (2014);

*This research was conducted while the first author held a summer internship in Microsoft Research, Redmond, and the last author was a Visiting Researcher there.

Yih et al. (2015)). In contrast, the synthetic questions in bAbI require rather complex reasoning over multiple computational steps while demanding only minimal semantic understanding. As the previous work on bAbI consists only of end-to-end models (Weston et al. (2014); Kumar et al. (2015); Sukhbaatar et al. (2015); Peng et al. (2015)), it is unclear whether incorrect answers arise from an imperfect semantic understanding, inadequate knowledge encoding, or insufficient model capacity (Dupoux (2015)). This is partly because the current paradigms based on neural networks have no interpretable intermediate representations which modelers can use to assess the knowledge present in the vectorial encoding of the system’s understanding of the input sentences. Our approach, in contrast, can illuminate what knowledge is captured in each representation via the formalism of TPR.

Tensor Product Representation (TPR), proposed by Smolensky (1990); Smolensky & Legendre (2006), is a mathematical method to represent complex structures from basic vectorial building blocks, so called *fillers* and *roles*. For example, one can encode a binary tree by *binding* filler vectors corresponding to the left- and right-child entities to role vectors corresponding to the ‘left child’ and ‘right child’ positions, respectively. Arbitrary trees can be represented by recursively applying the same method. As an outer product (i.e., tensor product) realizes the binding operation, both filler and role components are decodable from the resulting representation via the inner product; this is called *unbinding*. TPR is known to be capable of various applications such as tree operations, grammar processing and lambda-calculus evaluation (Smolensky (2012)).

In this paper, we endeavor to disentangle the problem cleanly into semantic parsing, knowledge encoding, and logical reasoning. Proposing two vector-space models inspired by TPR, we first provide an in-depth analysis of the bAbI dataset by clustering, based solely on their logical properties, the twenty question categories defined by bAbI. Such analysis enables us to conjecture why most existing models, in spite of their complexity, have failed to achieve good accuracy on *positional reasoning* and *path finding* tasks, whereas Peng et al. (2015) achieved successful results. If the bAbI tasks turn out to be considerably simpler than intended for its ultimate purpose of providing a major step towards “AI-complete question answering”, then more elaborated tasks will be required to test the power of proposed QA models such as memory networks.

As a further contribution, we also develop the foundation of a theory that maps inference for logical reasoning to computation over TPRs, generalizing our models under the rigorous TPR formalism. Due to the page limit, this theoretical foundation is relegated to the supplementary materials (Smolensky et al. (2016)). The experimental results show that accurate inference based on common-sense knowledge is transparently attainable in this formalism. We hope our exploration can contribute to the further improvement of end-to-end models toward the transparency and interpretability. To the best of our knowledge, our in-depth analysis of bAbI and of logical reasoning over distributed vectorial representations are each the first of their kind.

2 RELATED WORK

Since the seminal work of Bengio et al. (2003), researchers have paid increasing attention to various distributed representations in continuous vector spaces. In the computer science literature, Skipgram/CBoW (Mikolov et al. (2013)) and GloVe (Pennington et al. (2014)) are popular models that are trained based on the distributional similarities in word co-occurrence patterns; they have been frequently utilized as initial embeddings for a variety of other NLP tasks. In the cognitive science literature, on the other hand, BEAGLE (Jones & Mewhort (2007)) and DVRS (Ustun et al. (2014)) are trained differently, with random initializations and circular convolution. They assign two vectors for each word: an environmental vector to describe physical properties and a lexical vector to indicate meaning.

Whereas such representations are known to provide a useful way to incorporate prior linguistic knowledge, their usefulness is not clear for reasoning-oriented tasks. In other contexts, Grefenstette (2013) shows how to simulate predicate logic with matrices and tensors. Similarly, Rocktaschel et al. (2014) try to find low-dimensional embeddings which can model first-order logic in a vectorial manner. These models are only concentrated on general logic problems without considering NLP tasks. Note that vectorial encodings are necessary in many machine learning models such as neural networks. Reasoning based on linguistic cues in vector space uniquely characterizes our paper among these relevant work.

Category 1: Single Supporting Fact	Category 3: Three Supporting Facts
01: Mary moved to the bathroom.	01: Sandra went back to the hallway.
02: John went to the hallway.	02: Daniel took the apple.
03: Where is Mary? bathroom 1	03: John travelled to the kitchen.
04: Daniel went back to the hallway.	04: Daniel travelled to the bedroom.
05: Sandra moved to the garden.	05: Daniel got the football there.
06: Where is Daniel? hallway 4	06: Daniel went to the hallway.
	07: Where was the apple before the hallway? bedroom 2 6 4
	08: Mary went back to the bedroom.
	09: Daniel discarded the football.
	10: Daniel got the football.
	11: Mary went to the garden.
	12: Daniel travelled to the office.
	13: Daniel went back to the bedroom.
	14: Where was the football before the bedroom? office 10 13 12
	15: Daniel went back to the hallway.
	16: Mary went back to the bathroom.
	17: Daniel dropped the apple.
	18: Sandra journeyed to the kitchen.
	19: Where was the apple before the office? hallway 17 12 6

Figure 1: Sample statements(black), questions(blue), answers(red), and clues(green) for Categories 1, 2, and 3.

The tasks in bAbI have been studied mainly within the context of the Memory Network (MemNN) model, which consists of an array of representations called “memory” and four learnable modules: the I-module encodes the input into feature representation, the G-module updates relevant memory slots, the O-module performs inferences to compute output features given the input representation and the current memory, and finally the R-module decodes the output feature-based representation to the final response. Since the proposal of the basic MemNN (Weston et al. (2014)) model, the Adaptive/Nonlinear MemNN (Weston et al. (2015)), DMN (Kumar et al. (2015)), and MemN2N (Sukhbaatar et al. (2015)) models have been developed by varying certain parts of these modules. Nonetheless, none of these models except Peng et al. (2015) successfully accomplish either positional reasoning or path finding tasks. Our speculation about the performance by Peng et al. (2015) will be given in a later section based on our bAbI analysis.

3 MODELS AND ANALYSIS

The bAbI dataset consists of twenty different types of questions where each question category is claimed to be atomic and independent from the others (Weston et al. (2015)). In this section, we investigate clusters of categories with sample QA problems, analyzing what kinds of logical properties are shared across various types. We also elucidate, based on our vector space models, why it is difficult to achieve good accuracy on certain categories: positional reasoning and path finding.

3.1 CONTAINEE-CONTAINER RELATIONSHIP

Supporting Facts (1, 2, 3) The first three question categories of bAbI ask for the current or previous locations of actors and objects based on the statements given prior to the question. Category 1–3 questions respectively require precisely one, two, or three supporting facts to reason out the proper answers. Figure 1 illustrates sample statements and questions extracted from real examples in the training set. Reasoning in Category 1 implicitly requires a simple common-sense reasoning rule that “An actor cannot exist in two different locations at the same time.” In order to answer the questions in Category 2, we implicitly need another rule that “An object that belongs to an actor follows its owner’s location.” Further, if an item is dropped at one particular location, it will permanently stay in that location until someone grabs it and moves around with it later.

While two independent relations, *pick/drop* and *move*, seem to be involved in parallel in the Category 2 tasks, these questions can be all uniformly answered under the transitivity of a *containee belongs to a container*. If an actor moves to a location, he/she (a containee) now belongs to that location

(a container). Similarly, if an actor acquires an object, the item (a containee) newly belongs to that actor (a container). Transitivity then logically implies that the object belongs to the location occupied by the owner.

#	Statements/Questions	Relational Translations/Answers	Encodings/Clues
1	Mary went to the kitchen.	<i>Mary belongs to the kitchen (from nowhere).</i>	mk^T $m(k \circ n)^T$
3	Mary got the football there.	<i>The football belongs to Mary.</i>	fm^T fm^T
4	Mary travelled to the garden.	<i>Mary belongs to the garden (from the kitchen).</i>	mg^T $m(g \circ k)^T$
5	Where is the football?	garden	3, 4
9	Mary dropped the football.	<i>The football belongs to where Mary belongs to.</i>	fg^T fg^T
10	Mary journeyed to the kitchen.	<i>Mary belongs to the kitchen (from the garden).</i>	mk^T $m(k \circ g)^T$
11	Where is the football?	garden	9, 4

Table 1: Sample *containee-belongs-to-container* translations and corresponding encodings about Mary from Category 2. Symbols in encodings are all d -dimensional vectors for actors (*mary*), objects (*football*), and locations (*nowhere*, *kitchen*, *garden*). Translations and encodings for Category 3 are also specified with the parentheses and circle operation, respectively.

Knowing that every actor and object is unique without any ambiguity, one can encode such containee-container relationships by the following model using distributed representations. Assume all entities: actors, objects, and locations are represented by d -dimensional unit vectors in \mathbb{R}^d .¹ Then each statement is encoded by a second-order tensor (or matrix) in which the containee vector is bound to the container vector via the fundamental binding operation of TPR, the tensor (or outer) product² — in tensor notation, (*containee*) \otimes (*container*), or in matrix notation, (*containee*)(*container*)^T — and then stored in a slot in a memory. When an item is dropped, we perform an inference to store the appropriate knowledge in memory. For the example in Table 1, the container of the football at Statement 9 — the garden — is determined after figuring out the most recent owner of the football, Mary; transitivity is implemented through simple matrix multiplication of the encodings of Statement 3 (locating the football) and Statement 4 (locating the football’s current owner, Mary):

$$(fm^T) \cdot (mg^T) = f(m^T \cdot m)g^T = fg^T \quad (\because m^T m = \|m\|_2^2 = 1)$$

Finally, Category 3 asks the trajectory of items considering the previous locations of actors. Thus the overall task is to understand the relocation sequence of each actor and from this to reconstruct the trajectory of item locations. Whereas MemNNs introduced an additional vector for each statement for encoding a time stamp, we define another binding operation $\circ : \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}^d$. This binding operation maps a pair of (*next*, *prev*) location vectors into a d -dimensional vector via a $d \times 2d$ temporal encoding matrix U like the following:

$$n \circ p = U \begin{bmatrix} n \\ p \end{bmatrix} \in \mathbb{R}^d.$$

In Table 1, the second expression in the Encodings column specifies temporal encodings that identify location transitions: Statement 4, translated as *Mary belongs to the garden (from the kitchen)*, is encoded as $m(g \circ k)^T$. We can now reason to the proper answers for the questions in Figure 1 by the following inference steps, using basic encodings (for C1 & C2) and temporal encodings (for C3):

C1. Where is Mary?

- Left-multiply by m^T all statements prior to time 3. (Yields $m \cdot m^T b^T$, $m^T \cdot j h^T$.)
- Pick the most recent container where 2-norms of the multiplications in (a) are approximately 1.0. (Yields b^T ; $m^T j$ is small.)
- Answer by finding the location corresponding to the result representation. \Rightarrow **bathroom**

C2. Where is the football?

¹Topologically speaking, the unit hypersphere can be constructed by adding one more point (“at infinity”) to Euclidean space. Thus sampling from the hypersphere does not limit the generality of representations.

²In TPR terms, the containee corresponds to a filler, and the container corresponds to a role.

- (a) Left-multiply by f^T all statements prior to the current time. (Yields $f^T \cdot mk^T, f^T \cdot so^T, f^T \cdot fm^T, f^T \cdot mg^T$.)
- (b) Pick the most recent container where 2-norms of the multiplications in (a) are approximately 1.0. (Yields m^T .)
- (c) If the container is an actor (e.g., Mary in statement 3),
 - Find the most recent container of the actor by left-multiplying by m^T (Yields g^T .)
 - Answer by the most recent container. \Rightarrow **garden** for the questions at time 5 and 8.
- (d) If the container is a location (e.g., garden in statement 9), simply answer by the container.

C3. Where was the apple before the hallway?

- (a) Left-multiply by a^T all existing temporal encodings prior to time 7. (Yields $a^T \cdot s(h \circ n)^T, a^T \cdot ad^T, \dots$.)
- (b) Pick the earliest container (the start of the trajectory). \Rightarrow *Daniel* in statement 2.
- (c) Find the containers of Daniel by left-multiplying by d^T the temporal encodings between time 2 and 7. (Yields $d^T \cdot ad^T, d^T \cdot j(k \circ n)^T, d^T \cdot d(b \circ n)^T, d^T \cdot fd^T, d^T \cdot d(h \circ b), \dots$.)
- (d) By multiplying by the pseudo-inverse U^\dagger , unbind $2d$ -dimensional vectors between time 4 and 7. (Yields $U^\dagger(b \circ n) \approx [b; n]$, then $[h; b]$.)
- (e) Reconstruct the item trajectory in sequence. \Rightarrow *nowhere* \rightarrow *bedroom* \rightarrow *hallway*
- (f) Answer with (the most recent) location which is prior to the hallway. \Rightarrow **bedroom**

Three Argument Relations (5) In this category, there is a new type of statement which specifies ownership transfer: an actor gives an object to another actor. Since now some relations involve three arguments, (*source-actor, object, target-actor*), we need to encode an ownership trajectory instead of a location trajectory.

#	Statements/Questions	Relational Translations/Answers	Encodings/Clues
1	Jeff took the milk there.	<i>The milk belongs to Jeff (from None).</i>	$m(j * n)^T$
2	Jeff gave the milk to Bill.	<i>The milk belongs to Bill (from Jeff).</i>	$m(b * j)^T$
3	Who did Jeff give the milk to?	Bill	2
4	Daniel travelled to the office.	<i>Daniel belongs to the office.</i>	do^T
5	Daniel journeyed to the hallway.	<i>Daniel belongs to the hallway.</i>	dh^T
6	Who received the milk?	Bill	2
7	Bill went to the kitchen.	<i>Bill belongs to the kitchen.</i>	bk^T
8	Fred grabbed the apple there.	<i>The apple belongs to Fred (from none).</i>	$a(f * n)^T$
9	What did Jeff give to Bill?	milk	2

Table 2: Sample *containeer-belongs-to-container* translations and corresponding encodings for an example from Category 5. Symbols in encodings are all d -dimensional vectors for actors (*nobody, jeff, daniel, bill, fred*), objects (*milk, apple*), and locations (*office, kitchen*).³

Analogously to the \circ operation used for Category 3, we realize the $*$ operation by defining a map $*$: $\mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}^d$. This new binding operation maps a pair of (*next, prev*) owner vectors into a d -dimensional vector via a $d \times 2d$ matrix V in the exactly same fashion: $n * p = V[n; p] \in \mathbb{R}^d$. Due to the similarity in encoding, the inference is also analogous to the inference for Category 3.

C5. Three questions of Table 2?

- (a) Find the owners of the milk by left-multiplying by m^T the encodings prior to time 3.
- (b) Unbind the owner transitions by multiplying them by the pseudo-inverse V^\dagger .
- (c) Reconstruct the ownership trajectory for the milk. \Rightarrow *Nobody* \rightarrow *Jeff* \rightarrow *Bill*
- (d) Answer accordingly each question based on the trajectory.

Though no more complex examples or distinct categories exist in the dataset, it is clear that our encoding scheme is capable of inferring the full trajectory of item location considering both relocation of actors and transfers of ownership. In such cases, both \circ and $*$ will be used at the same time in

³To avoid notational confusion, we modify the name of an actor (from Mary to Daniel) and a location (from the bathroom to the office) from the real example in Category 5.

<p>Category 6: Yes/No Questions</p> <p>01: Daniel went back to the hallway.</p> <p>02: John got the apple there.</p> <p>03: Is Daniel in the hallway? yes 1</p> <p>04: John dropped the apple.</p> <p>05: Mary got the apple there.</p> <p>06: Is Daniel in the hallway? yes 1</p> <p>07: Daniel moved to the bedroom.</p> <p>08: Sandra travelled to the hallway.</p> <p>09: Is Daniel in the hallway? no 7</p>	<p>Category 7: Counting</p> <p>01: Mary took the apple there.</p> <p>02: John travelled to the office.</p> <p>03: How many objects is Mary carrying? one 1</p> <p>04: Mary travelled to the bathroom.</p> <p>05: Sandra went back to the bedroom.</p> <p>06: How many objects is Mary carrying? one 1</p> <p>07: Mary got the football there.</p> <p>08: Mary went to the office.</p> <p>09: How many objects is Mary carrying? two 1 7</p> <p>10: Mary passed the apple to John.</p> <p>11: Mary left the football.</p> <p>12: How many objects is Mary carrying? none 1 7 10 11</p>
<p>Category 8: List/Sets</p> <p>01: Mary took the milk there.</p> <p>02: Mary went to the office.</p> <p>03: What is Mary carrying? milk 1</p> <p>04: Mary took the apple there.</p> <p>05: Sandra journeyed to the bedroom.</p> <p>06: What is Mary carrying? milk,apple 1 4</p> <p>07: Mary put down the milk.</p> <p>08: Mary discarded the apple.</p> <p>09: What is Mary carrying? nothing 1 7 4 8</p>	<p>Category 9: Simple Negation</p> <p>01: Sandra travelled to the garden.</p> <p>02: Sandra is no longer in the garden.</p> <p>03: Is Sandra in the garden? no 2</p> <p>04: Sandra is in the garden.</p> <p>05: Sandra journeyed to the hallway.</p> <p>06: Is Sandra in the hallway? yes 5</p>

Figure 2: Sample statements(black), questions(blue), answers(red), and clues(green) for Category 6, 7, 8, and 9. Answer types are different from the previous categories.

encoding. (e.g., encoding for time 5 will be then $d(h \circ o)^T$). Note also that there may be multiple transfers between the same pair of actors in a history prior to the given question. While any of them could be appropriate evidence to justify different answers, the ground-truth answers in the training set turned out to be all based on the most recent clues.

Answer Variations (6, 7, 8, 9) As shown in Figure 2, the responses to questions of Categories 6-9 require different measures of the inferred element. For example, the statements in Category 6 are structurally equivalent to the statements in Category 2, while the questions concern only a current location, similar to Category 1. However, each question is formulated in a binary yes/no format, confirming “*Is Daniel in the hallway?*” instead of asking “*Where is Daniel?*”. Category 7 is isomorphic to Category 5 in the sense that actors can pick up, drop, and pass objects to other actors. However, each question inquires the number of objects currently belonging to the given actor. On the other hand, a response in Category 8 must give the actual names of objects instead of counting their number. The statements in this category are based not on Category 5, but on Category 2 which is simpler due to the lack of ownership transfer. Lastly, statements in Category 9 can contain a negative quantifier such as ‘*no*’ or ‘*no longer*’. Responses confirm or not the location of actors via yes/no dichotomy as for Category 6. However, the overall story is based on the simplest Category 1.

Since answer measures are the only differences of these categories from Category 1, 2, 3, and 5, no additional encodings or inferences are necessary. However, there are several caveats in formulating actual answers: 1) For yes/no questions, we should know the answers must be either *yes* or *no* in advance based on the training examples. 2) When counting the number of belongings, the answer must use English number words rather than Arabic numerals. 3) When enumerating the names of belongings, names must be sequenced by their order of acquisition. 4) A negative quantifier is realized by binding the initial default location *nowhere* back to the given actor. Note that there is no double negation.

Statement Variations (10, 11, 12, 13) Statements in Categories 10-13 contain more challenging linguistic elements such as conjunctions (*and/or*) or pronouns (*he/she/they*). While statements in Category 10 is structurally similar to Category 1’s, an actor can be located in *either* one *or* another location. Due to such uncertainty, some questions must be answered indefinitely by ‘*maybe*’. On the other hand, each statement in Category 12 can contain multiple actors conjoined by ‘*and*’ to indicate that these actors all carry out the action. Aside from such conjunctions, statements and questions are isomorphic to Category 1’s. Statements in Categories 11/13 can consist of a singular/plural pronoun to indicate single/multiple actors mentioned earlier. Since coreference resolution is itself a

<p>Category 10: Indefinite Knowledge</p> <p>01: Julie travelled to the kitchen. 02: Bill is either in the school or the office. 03: Is Bill in the office? maybe 2 04: Bill went back to the bedroom. 05: Bill travelled to the kitchen. 06: Is Bill in the kitchen? yes 5</p>	<p>Category 11: Basic Coreference</p> <p>01: Mary went back to the bathroom. 02: After that she went to the bedroom. 03: Where is Mary? bedroom 1 2 04: Daniel moved to the office. 05: Afterwards he moved to the hallway. 06: Where is Daniel? hallway 4 5</p>
<p>Category 12: Conjunction</p> <p>01: Daniel and Sandra went back to the kitchen. 02: Daniel and John went back to the hallway. 03: Where is Daniel? hallway 2 04: Daniel and John moved to the bathroom. 05: Sandra and Mary travelled to the office. 06: Where is Daniel? bathroom 4</p>	<p>Category 13: Compound Coreference</p> <p>01: Mary and Daniel went to the bathroom. 02: Then they journeyed to the hallway. 03: Where is Daniel? hallway 1 2 04: Sandra and John moved to the kitchen. 05: Then they moved to the hallway. 06: Where is John? hallway 4 5</p>

Figure 3: Sample statements(black), questions(blue), answers(red), and clues(green) for Category 10, 11, 12, and 13. Statement types are different from the previous categories.

difficult problem, all pronouns are limited to refer only to actors mentioned in the immediately prior statement.

To encode conjunctions, we can still leverage the same method: conjoin two objects by another bilinear binding operation $\star : \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}^d$, and unbind similarly via the pseudo-inverse of the corresponding matrix. In our implementation, every statement is encoded using such a binding operation. For instance, the first two statements of the given Category 10 example are encoded into $j(k \star k)^T$ and $b(s \star o)^T$, with \star encoding *or*. If two locations unbound from the target actor are identical, we output a yes/no definite answer, whereas two different locations imply the indefinite answer ‘*maybe*’ if one of the unbound locations matches the queried location. For the conjunction *and* in Category 12, exactly the same formalism is applicable for conjoining actors instead. Whereas a singular pronoun appearing at time t in Category 11 is simply replaced by the actor mentioned at time $t - 1$, we also use \star -binding to provide the multiple coreference needed for Category 13. For instance, the first statement in the given Category 13 example is encoded as $(m \star d)b^T$ and the same encoding is substituted for ‘*they*’ to represent the actors in the following statement.

Deduction/Induction (15, 16, 18, 20) While the statements and questions in these categories seem different at first glance, their goals are all to reason using a transitivity-like rule. Categories 15 creates a food chain among various animals, and Category 18 yields a partial/total order of sizes among various objects. Whereas inference in these two categories is deductive, Categories 16 and 20 require inductive inference. In all four categories, every statement is easily represented by a containee-container relation obeying transitivity. For instance, the Category 15 example of Figure 4 is encoded by $\{mc^T, wm^T, cs^T, sw^T\}$. Then the answer for the first question: “*What is Jessica afraid of?*” will be answered by left-multiplying these by the transpose of $j = m$ and finding the one whose norm is approximately 1.0, which is mc^T . Thus the result $j^T \cdot (mc^T) = m^T(mc^T) = (m^T m)c^T = c^T$ produces the desired answer *cat*. Similarly, in Category 18, if question encoding (e.g., “*Does the chocolate fit in the box?*” = cb^T) is achievable by some inner products of statement encodings, the answer must be ‘*yes*’, otherwise, ‘*no*’.

On the other hand, in Category 16, transitivity is applied reversely as a *container-containee* fashion. For instance, “*Lily is a lion*” is encoded by ℓl^T , whereas “*Lily is green*” is encoded by lg^T . In encoding “*x is-a Y*”, we put the more general concept at the left side of the outer-product binding Yx^T ; to encode “*x has-property Z*” we use xZ^T . This allows us to induce a property for the general category *Y* based on the single observation of one of its members, via simple matrix multiplication, just as transitive inference was implemented above: $(\ell l^T) \cdot (lg^T) = lg^T$, meaning “*lion is green.*” Similarly in Category 20, there exists precisely one statement which describes a property of an actor (e.g., “*Sumit is bored.*” = bs^T). Then a statement describes the actor’s relocation (e.g., “*Sumit journeyed to the garden.*” = sg^T), yielding an inductive conclusion by matrix multiplication: “*Being boring makes people go to the garden.*” = $(bs^T) \cdot (sg^T) = bg^T$. The inductive reasoning also generalizes to other actions (e.g., the reason for later activity, “*Sumit grabbed the football.*” = sf^T , is also being bored, because $(bs^T) \cdot (sf^T) = bf^T$).

<p>Category 15: Basic Deduction</p> <p>01: Mice are afraid of cats. 02: Emily is a mouse. 03: Wolves are afraid of mice. 04: Cats are afraid of sheep. 05: Winona is a cat. 06: Sheep are afraid of wolves. 07: Jessica is a mouse. 08: Gertrude is a sheep. 09: What is Jessica afraid of? cat 7 1 10: What is Emily afraid of? cat 2 1 11: What is Jessica afraid of? cat 7 1 12: What is Winona afraid of? sheep 5 4</p>	<p>Category 18: Reasoning about Size</p> <p>01: The suitcase is bigger than the container. 02: The container fits inside the box. 03: The chest is bigger than the chocolate. 04: The suitcase fits inside the box. 05: The chest fits inside the box. 06: Does the chocolate fit in the box? yes 5 3 07: Does the chocolate fit in the box? yes 5 3 08: Does the box fit in the container? no 1 4 09: Is the box bigger than the chocolate? yes 5 3 10: Does the box fit in the chocolate? no 3 5</p>
<p>Category 16: Basic Induction</p> <p>01: Bernhard is a lion. 02: Julius is a lion. 03: Lily is a lion. 04: Bernhard is green. 05: Lily is green. 06: Brian is a lion. 07: Greg is a swan. 08: Greg is gray. 09: Julius is yellow. 10: What color is Brian? green 6 3 5</p>	<p>Category 20: Reasoning about Motivations</p> <p>01: Sumit is bored. 02: Where will Sumit go? garden 1 03: Yann is hungry. 04: Where will Yann go? kitchen 3 05: Yann went back to the kitchen. 06: Why did Yann go to the kitchen? hungry 3 07: Sumit journeyed to the garden. 08: Why did Sumit go to the garden? bored 1 09: Yann picked up the apple there. 10: Why did Yann get the apple? hungry 3 11: Sumit grabbed the football there. 12: Why did Sumit get the football? bored 1</p>

Figure 4: Sample statements(black), questions(blue), answers(red), and clues(green) for Category 15, 16, 18, and 20. Categories 15 and 18 create chains from smaller/weaker to stronger/larger, whereas Categories 16 and 20 from general ones to specific ones.

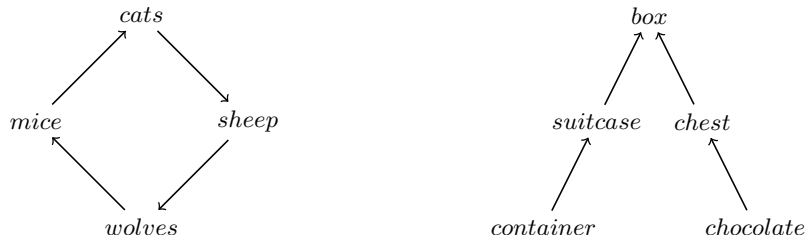


Figure 5: The circular food chain (Category 16) and the partial order (Category 18) corresponding to the examples in Figure 4. The arrows imply *afraid-of* and *fits-inside* relations, respectively.

Prior Knowledge (4, 14) Though statements in Category 4 looks quite dissimilar from those in the other categories, they can be eventually modeled by a uni-relational reasoning chain based on the containee-container relation, provided we know that ‘north’ and ‘south’ are opposite to each other. Thus the first two statements in the first Category 4 example in Figure 6 yield $\{ko^T, gk^T\}$, from which we infer $(gk^T) \cdot (ko^T) = go^T$ “The office is north of the garden.” While the questions are all simple knowledge confirmation, note that a relational word (e.g., ‘east’) might never appear in the prior statements, as illustrated in the second example of Category 4 in Figure 6. However the most important point is that two non-collinear relations (e.g., ‘north’, ‘east’) never appear together in the same example.

On the other hand, statements in Category 14 are no longer chronologically ordered. In order to infer a correct locational trajectory without repeating statements multiple times, we predefine four vectors for each time stamp: *yesterday*, *this morning*, *this afternoon*, and *this evening*, and bind location with the corresponding stamp instead of the previous location. For example, the encoding for the statement at time 2 now becomes $j(b \circ m)^T$ instead of $j(b \circ p)^T$. Knowing the correct order of these four time stamps, which could be learned from the training examples, we can easily reorder by unbinding time stamps.

<p>Category 4. Two Argument Relation</p> <p>01: The office is north of the kitchen. 02: The garden is south of the kitchen. 03: What is north of the kitchen? office 1</p> <hr/> <p>01: The kitchen is west of the garden. 02: The hallway is west of the kitchen. 03: What is the garden east of? kitchen 1</p>	<p>Category 17: Positional Reasoning</p> <p>01: The triangle is above the pink rectangle. 02: The blue square is to the left of the triangle. 03: Is the pink rectangle to the right of the blue square? yes 1 2</p> <hr/> <p>01: The red sphere is below the yellow square. 02: The red sphere is above the blue square. 03: Is the blue square below the yellow square? yes 2 1</p>
<p>Category 14: Time Manipulation</p> <p>01: Yesterday Julie went back to the park. 02: Julie went to the bedroom this morning. 03: Bill journeyed to the cinema yesterday. 04: This morning Bill went back to the park. 05: Where was Bill before the park? cinema 4 3 06: This evening Julie went to the school. 07: This afternoon Julie went back to the park. 08: Where was Julie before the bedroom? park 2 1</p>	<p>Category 19: Path Finding</p> <p>01: The bedroom is south of the hallway. 02: The bathroom is east of the office. 03: The kitchen is west of the garden. 04: The garden is south of the office. 05: The office is south of the bedroom. 06: How do you go from the garden to the bedroom? n,n 4 5</p>

Figure 6: Sample statements(black), questions(blue), answers(red), and clues(green) for Category 4, 14, 17, and 19. Categories 4 and 17 contains two different examples separated by a horizontal line.

3.2 MULTIPLE RELATIONSHIPS

Path Finding (19) Our goal in this category is to find the path from one location to another location in a Manhattan-grid-like sense. Note that if A is north of B , and B is north of C , then the right path from A to C in grid must be ‘north, north’ rather than simply ‘north’. We assume given four $d \times d$ non-singular matrices N, E, W, S encoding four different directions satisfying $N = S^{-1}$ and $E = W^{-1}$. Then

#	Statements/Questions	Translations/Answers/Clues	Encodings	Seq
1	The bedroom is south of the hallway.	Decides b given the initial h .	$b = Sh$	(1)
2	The β athroom is east of the office.	Defer until we know either o or β .	$\beta = Eo$	(3)
3	The kitchen is west of the garden.	Defer until we know either g or k .	$k = Wg$	(5)
4	The garden is south of the office.	Defer until we know either o or g .	$g = So$	(4)
5	The office is south of the bedroom.	Decides o given b .	$o = Sb$	(2)
6	How do you go from the garden to the bedroom?	n,n 4, 5	$b = Xg$	(6)

Table 3: Sample multi-relational translations and corresponding encodings from Category 19. Symbols in encodings are either d -dimensional object vectors (hallway, bedroom, office, β athroom, garden, kitchen) or $d \times d$ directional matrices (South, East, West, North). The last column shows the sequence of actual running order.

After initializing the first object in the right-hand side (e.g., ‘hallway’) by a random vector, we decide the rest of the object vectors in sequence by multiplying the directional matrix (or its inverse in case that the right-hand side is unknown and the left-hand side is known). In case that both sides are unknown, we defer such a statement by putting it into a queue. In fact, the solution path X can be determined either by selecting, of all combinations of two directions $\{NN, NE, NW, NS, \dots, SN, SE, SW, SS\}$, the one which best satisfies $b = Xg$ (in the example of Table 3) or by solving this equation based on iterative substitutions. Note also that we need to know that (n, e, w, s) in the answers correspond to (north, east, west, south), respectively, which could be learned from training data.

Positional Reasoning (17) While this category could be seen similar to Path Finding, each question only asks a relative position between two objects. For instance, if “ r is below s ”, and “ b is below r ”, then the position of b with respect to s must be simply ‘below’ rather than ‘below, below’. Even if an object is mentioned to be left of another object, it could be also located in left-above or left-below of another object. Due to these subtleties, we here adopt redundant representations with four $d \times d$ singular matrices (A, B, L, R) corresponding to four directions: (above, below, left, right).

For this directional subsumption, in contrast to the non-singularity of the directional matrices for Category 19, we now strictly enforce idempotency to these matrices (i.e., $A^n = \dots = A^2 = A \neq I$). Then we define the following four $4d \times 4d$ block matrices and encode each statement with these matrices in the same manner as for Category 19.

$$\mathcal{A} = \begin{bmatrix} A & 0 & 0 & 0 \\ 0 & I & 0 & 0 \\ 0 & 0 & I & 0 \\ 0 & 0 & 0 & I \end{bmatrix} \quad \mathcal{B} = \begin{bmatrix} I & 0 & 0 & 0 \\ 0 & B & 0 & 0 \\ 0 & 0 & I & 0 \\ 0 & 0 & 0 & I \end{bmatrix} \quad \mathcal{L} = \begin{bmatrix} I & 0 & 0 & 0 \\ 0 & I & 0 & 0 \\ 0 & 0 & L & 0 \\ 0 & 0 & 0 & I \end{bmatrix} \quad \mathcal{R} = \begin{bmatrix} I & 0 & 0 & 0 \\ 0 & I & 0 & 0 \\ 0 & 0 & I & 0 \\ 0 & 0 & 0 & R \end{bmatrix}$$

In this encoding, each of the four d -dimensional subspaces of \mathbb{R}^{4d} plays a role of indicating relative positions with respect to (above, below, left, right), independently. Carrying out the encoding of “ r is below s ”, $r = \mathcal{B}s$, ensures that the components of r and s differ only in the dimensions from $(d + 1)$ to $2d$ (from the B block of \mathcal{B}); that is, $r_k = s_k$ for $k = 1, 3, 4$ (where s_i indicates the i -th d -dimensional sub-block of s). This is actually inconsistent with the encoding of “ s is above r ”, which demands that s and r differ only in their first sub-block. Thus in order to determine whether or not s is indeed above r , it is necessary to check whether $r_2 = \mathcal{B}s_2$ as well as whether $s_1 = \mathcal{A}r_1$. If either condition is satisfied, we can confirm ‘ s is above to r ’. Similarly, horizontal relations must be checked on both the third and fourth d -dimensional sub-blocks.

4 EXPERIMENTAL RESULTS

We implement our models and algorithms under the analysis given in the previous section. Due to the small vocabulary (mostly less than or equal to four elements among actors, objects, locations, and actions) and non-ambiguous grammars, a simple dependency parser⁴ and basic named entity recognition enable us to achieve 100% accurate semantic parsing. Then we translate every statement into a representation based on the appropriate containee-container or multiway relation, and then store it in an array of memory slots. The logical reasoning after semantic parsing and knowledge representation no longer refers to the original text symbols.

Type	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10
Accuracy	100%	100%	100%	100%	99.3%	100%	96.9%	96.5%	100%	99%
Model	MNN	MNN	MNN	MNN	DMN	MNN	DMN	DMN	DMN	SSVM
Type	C11	C12	C13	C14	C15	C16	C17	C18	C19	C20
Accuracy	100%	100%	100%	100%	100%	100%	72%	95%	36%	100%
Model	MNN	MNN	MNN	DMN	MNN	MNN	Multitask	MNN	MNN	MNN

Table 4: Best accuracies for each category and the model that achieved the best accuracy. MNN indicates Strongly-Supervised MemNN trained with the clue numbers, and DMN indicates Dynamic MemNN, and SSVM indicates Structured SVM with the coreference resolution and SRL features. Multitask indicates multitask training.

In contrast to all previous models reported in Table 4, in Table 5 we also report test accuracy on the training data to measure how well our models incorporate common sense. Note that testing on the training data is available because our training procedure only parses the appropriate semantic components such as actors, objects, locations, actions, and the forms of answers without using given answers and clues for tuning the model parameters.

Note that the imperfect accuracy in Category 16 is due to the ambiguity of evidence. As given in Figure 4, one can answer the color of *Brian* as ‘yellow’ because the latest evidence tells *Julius who is a lion is yellow*. Similarly, in Category 5, the 8th story consists of incorrect/inconsistent answers at time 14 and 17 (for training), as they ignore the most recent ownership transfers and choose some old history as ground-truth answers. (The 63rd and 186th stories in the test data also consist of incorrect answers, at time 27 and 22, respectively) Other than these two categories, we achieve perfect accuracies performing common-sense operations only on representations in memory.

⁴We use Stanford Dependency Parser. <http://nlp.stanford.edu/software/stanford-dependencies.shtml>

Type	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10
Training	100%	100%	100%	100%	99.8%	100%	100%	100%	100%	100%
Test	100%	100%	100%	100%	99.8%	100%	100%	100%	100%	100%
Type	C11	C12	C13	C14	C15	C16	C17	C18	C19	C20
Training	100%	100%	100%	100%	100%	99.4%	100%	100%	100%	100%
Test	100%	100%	100%	100%	100%	99.5%	100%	100%	100%	100%

Table 5: Accuracies on training and test data on our models. We achieve near-perfect accuracy in almost every category including positional reasoning and path finding.

As the experimental results show, there is a clear distinction between two sets of tasks. Tasks in most categories can be modeled by a containee-container-like relationship respecting a transitivity-like inference rule, whose goals are to create a linear/circular chain. On the other hand, positional reasoning and path finding require multiple relationships where each corresponding pair (e.g., north vs. south) has its own transitivity structure, operating independently of other pairs (e.g. east vs. west). We hypothesize that this difference poses a major difficulty for most of Memory Network models to perform an accurate inference for positional reasoning and path finding.

Recently, Neural Reasoner (NR) by Peng et al. (2015) improves the accuracy for these two difficult categories by a large margin, achieving 97.9% and 87.0% when using 10k training set.⁵ Different from other memory network models, NR has multiple reasoning layers. Starting from the initial statements and questions, NR constructs new statements and questions at the next layer, and repeats this process recursively over multiple layers. As both positional reasoning and path finding require generating inferences from, and new versions of, relevant statements for each relationship (e.g., “x is north of ” can become “y is south of x”), the abilities to generate new facts and to derive final answers by integrating them from multiple relationships could be a key reason why NR is successful, like our TPR-based reasoner. While NR in experiment is simplified so that all new facts maintain the same initial representations, the question representation changes for each layer considering all existing facts and the previously evolved question. Due to the simplicity of the task, we conjecture that evolving representations of the question could be sufficient to comprise the key ingredient for each multi-relationship. However, it seems that training such multiple layers requires a large amount of training data, yielding drastically different performance of NR on two different dataset sizes.

5 CONCLUSION

The major contributions of this paper are two-fold. First, we thoroughly analyze the recently acclaimed bAbI question-answering tasks by grouping the twenty categories based on their relational properties. Our analysis reveals that most categories except positional reasoning and path finding are governed by uni-relational characteristics. As these turn out to support inference in a similar manner under transitivity, it could be dangerous to evaluate the capacity of network models based only on their performance on bAbI. In contrast, two more difficult categories require the capability of performing multi-relational reasoning, a capability which is apparently missing in most previous models. One could later develop a more sophisticated dataset that needs substantially harder reasoning by introducing multiple relationships. Second, we propose two vector space models which can perform logistic reasoning for QA with distributed representations. While TPR has been used for various problems such as tree/grammar encoding and lambda-calculus evaluation, logical reasoning is a new area of application that requires iterative processing of TPRs. In subsequent work, we will generalize the vector-space approach for multi-relational problems. We hope these studies shed light on the viability of developing further reasoning models which can perform inference with existing knowledge in an interpretable and transparent manner.

⁵All accuracy values of various models reported in the experimental section of the present paper are based on a 1k training set. Neural Reasoner achieves 66.4% and 17.3% when using the 1k dataset.

REFERENCES

- Bengio, Yoshua, Ducharme, Rejean, Vincent, Pascal, and Janvin, Christian. A neural probabilistic language model. *JMLR*, 3:1137–1155, 2003.
- Berant, Jonathan, Chou, Andrew, Frostig, Roy, and Liang, Percy. Semantic parsing on Freebase from question-answer pairs. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pp. 1533–1544, Seattle, Washington, USA, October 2013. Association for Computational Linguistics. URL <http://www.aclweb.org/anthology/D13-1160>.
- Bordes, Antoine, Chopra, Sumit, and Weston, Jason. Question answering with subgraph embeddings. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 615–620, Doha, Qatar, October 2014. Association for Computational Linguistics. URL <http://www.aclweb.org/anthology/D14-1067>.
- Dupoux, Emmanuel. Deconstructing ai-complete question-answering: going beyond toy tasks. 2015. URL <http://bootphon.blogspot.com/2015/04/deconstructing-ai-complete-question.html>.
- Grefenstette, Edward. Towards a formal distributional semantics: Simulating logical calculi with tensors. Association for Computational Linguistics, 2013.
- Jones, Michael N. and Mewhort, Douglas J. K. Representing word meaning and order information in a composite holographic lexicon. *Psychological Review*, 114:1–37, 2007.
- Kumar, Ankit, Irsoy, Ozan, Su, Jonathan, Bradbury, James, English, Robert, Pierce, Brian, Ondruska, Peter, Gulrajani, Ishaan, and Socher, Richard. Ask me anything: Dynamic memory networks for natural language processing. *CoRR*, abs/1506.07285, 2015. URL <http://arxiv.org/abs/1506.07285>.
- Mikolov, Tomas, Sutskever, Ilya, Chen, Kai, Corrado, Greg S, and Dean, Jeff. Distributed representations of words and phrases and their compositionality. pp. 3111–3119. 2013.
- Peng, Baolin, Lu, Zhengdong, Li, Hang, and Wong, Kam-Fai. Towards neural network-based reasoning. *CoRR*, abs/1508.05508, 2015. URL <http://arxiv.org/abs/1508.05508>.
- Pennington, Jeffrey, Socher, Richard, and Manning, Christopher D. Glove: Global vectors for word representation. pp. 1532–1543, 2014. URL <http://www.aclweb.org/anthology/D14-1162>.
- Rocktaschel, Tim, Singh, Sameer, Bosnjak, Matko, and Riedel, Sebastian. Low-dimensional embeddings of logic. 2014.
- Smolensky, Paul. Tensor product variable binding and the representation of symbolic structures in connectionist systems. *Artificial Intelligence*, 46(1-2), 1990.
- Smolensky, Paul. Symbolic functions from neural computation. *Philosophical Transactions of the Royal Society*, 370:3543–3569, 2012.
- Smolensky, Paul and Legendre, Geraldine. *The Harmonic Mind: From Neural Computation to Optimality-Theoretic Grammar Volume I: Cognitive Architecture*. The MIT Press, 2006.
- Smolensky, Paul, Lee, Moontae, He, Xiaodong, Yih, Wen-tau, Gao, Jianfeng, and Deng, Li. Basic reasoning with tensor product representations. *Technical Report, Microsoft Research*, 2016. URL <http://arxiv.org/abs/1601.02745>.
- Sukhbaatar, Sainbayar, Szlam, Arthur, Weston, Jason, and Fergus, Rob. End-to-end memory networks. *CoRR*, abs/1503.08895, 2015. URL <http://arxiv.org/abs/1503.08895>.
- Ustun, Volkan, Rosenbloom, Paul S., Sagae, Kenji, and Demski, Abram. Distributed vector representations of words in the sigma cognitive architecture. 2014.
- Weston, Jason, Chopra, Sumit, and Bordes, Antoine. Memory networks. *CoRR*, abs/1410.3916, 2014. URL <http://arxiv.org/abs/1410.3916>.
- Weston, Jason, Bordes, Antoine, Chopra, Sumit, Mikolov, Tomas, Rush, Alexander M., and van Merriënboer, Bart. Towards ai-complete question answering: A set of prerequisite toy tasks. volume abs/1502.05698. 2015. URL <http://arxiv.org/abs/1502.05698>.
- Yih, Wen-Tau, Chang, MingWei, He, Xiaodong, and Gao, Jianfeng. Semantic parsing via staged query graph generation: Question answering with knowledge base. 2015.