Method

Data

Representing knowledge as triplets

We represent pieces of knowledge as triplets with . ....**Tobe finished**

Simple species data

We first extracted a data table of 8 species, with 36 relations each. This table is extracted from Appendix B, table B-1 in [citation, R&M book].

Negative sample generation

For each head-relation pair (h-r), we generate negative samples by appending false tails (t_f) that: (1) < h-r-t_f> is not in the dataset and (2) <h'-r-t_f> is in the dataset (in other words, r-t_f is a valid relation-tail pair). The ratio between negative and positive pairs is reportedly influential on training performance [cite: ?? multidist model or complex vector]. Here we tentatively set it to at most 3 negative samples per correct triplet (some triplets have less because not enough legitimate t_f .

Model

Reimplementing the classic PDP model

Network and parameters: We implemented the Rumelhart semantic memory model as shown in Figure 2.2 [citation R &M; could insert figure]. Specifically, head, relation and tails are represented in a one-hot manner; head is connected to a representation layer of same number of units as the head layer; then both the representation layer and relation connect to a hidden layer with 15 units, which feedforward to the output tail layer. Each layer's output is rectified with ReLU function (instead of softmax as in the original model), but we didn't test whether this is a significant modification. Also the output layer had a softmax to constrain the output between 0 and 1.

Training: We used pytorch package to construct this model. RMSprop was used for optimizing the network. 300 episodes are included in the training data.

Modern embedding model

Network and parameters: One recent popular trend of knowledge base model is the embedding model started by transE [citation Bordes et al 2013], where the head, relation and tail are transformed into an embedding vector space, and the knowledge triplets correspond to some linear algebra operations. [cite....] summarized some possible vector and operation choices. Here we only explored a specific form, "DistMulti model" [cite...], where the entities (both head and tails) are represented as vector while the relations is a diagonal matrix. A triplet is represented as a bilinear transformation"

$$Score(h - r - t) = v_h B_r v_t$$

For a correct triplet the score should be 1 while false knowledge would have a score of -1.

Specifically, we used a somewhat arbitrary embedding of 15 dimensions for both datasets.

Training: We used pytorch package to construct this model. Adam was used for optimizing the network. 1500 episodes are included in the training data.

Results

Small biology dataset

Here we first replicate the results shown in [R&M], then observe whether the embedding model show similar behaviors. Moreover, we introduced an inference task that has rarely been studied in the current knowledge base literature and developed algorithms to solve these tasks for the embedding model.

1. Learning process

First, we see both model could learn to distinguish the correct and wrong triplets. For example, we test the model output for triplets related with canary, adding a false triplet ("canary-is-red") as comparison:
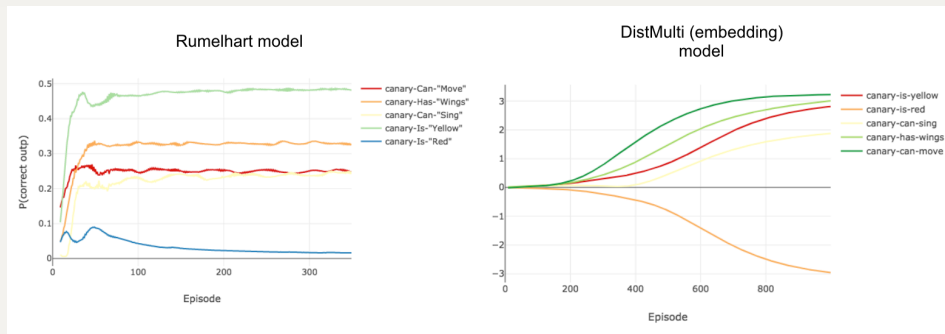
[to add: averaged across 10 runs.]

R&M mentioned an over-generation phenomenon in this dataset: "pine-has-leaves" might be memorized as correct because the other tree, oak, does have leaves. This overgeneralization has been observed in the embedding model as well:

The model at first thought that pine doesn't have leaves.

2. Emergence of clusters

To check whether similar entities will gradually gain more similar embeddings and evolve some hierarchical structure, we compare the correlation matrix of embeddings from the first episode to 1100 episode.
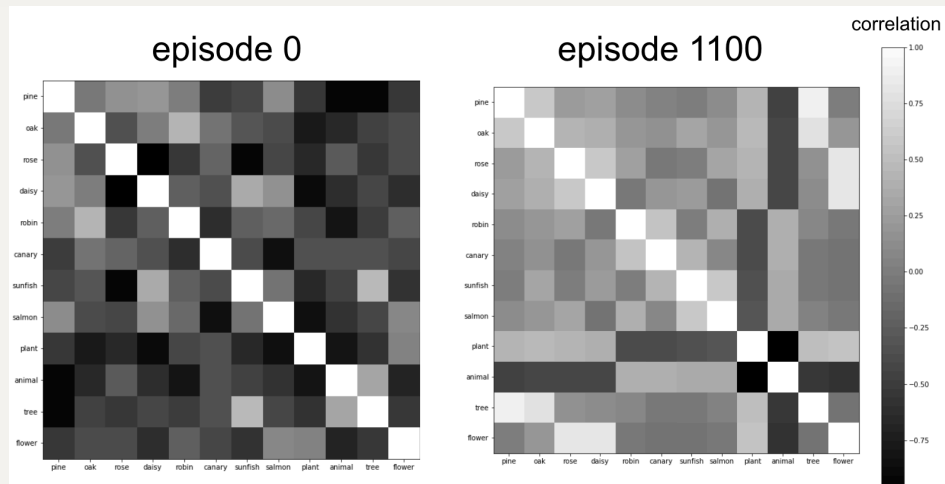
We can see the clustering structure emerging among the eight living things. Moreover, higher-order concepts like plants and animals show strong dissimilarity from each other as the common knowledge would reason. Note that the training data don't include triplets describing the abstract concept "animal" — it's only included in statements as "fish-isA-animal". Yet the model could learn to infer about the properties of animals. In the next section we more explicitly explore the inference from the embedding model.

3. Inference task

To examine what the model can learn about the abstract concepts that only appeared in the tails of "isA" relations, we picked some testing triplets using them as heads (thus not included in the training set) and checked the learning progress.

The embedding model turns out to be able to learn to assign correct judgements to these relations, e.g. knowing that "bird-isa-animal" is correct but "plant-isa-animal" is wrong.

4. Automated discovery

The novel triplets are given by human and approved by the model. Can our model discover novel knowledge by itself? Turns out we can simply search all the grammatical relations given an entity, and set a threshold hold to filter strong positive / negative judgements. With a threshold of score 4.5, our model learns true triplets such as "animal-has-skin", "flower-is-pretty", and negative samples as "animal-has-roots". Though this method generates many promising good triplets, the model also thinks "livingthing-can-grow" must be false, possibly because of the speciality in the "livingthing" entity itself.

Bigger Dataset

1. Learning process

Our larger dataset includes 766 entities, many are very specific such as "baby long-tailed weasel", "female little brown bat", etc. Thus it's hard to pick representative triplets to demonstrate the training process, and we simply checked the correctness of triplets:
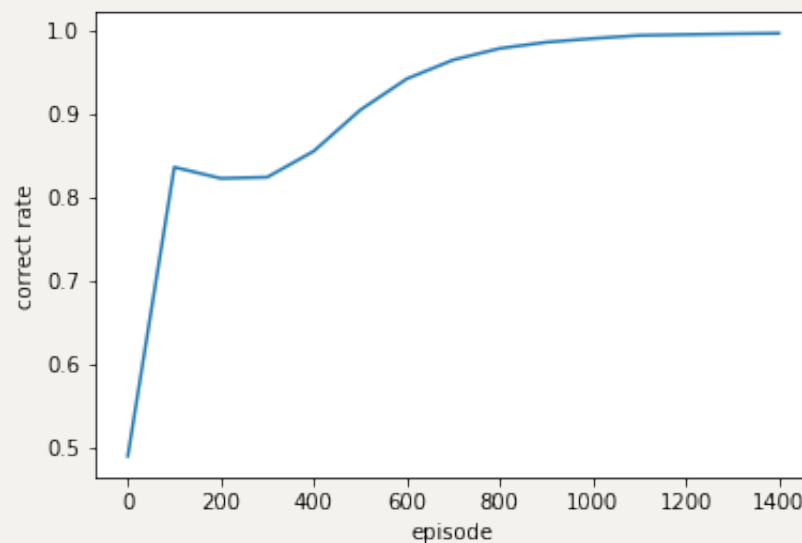
As is seen the model can indeed reach 100% correctness after around 1.2k episodes. Note we didn't even adjust the network or learning parameters from the small dataset. It's unclear to us why is there a dip in correctness at around episode 300.

2. Emergence of clusters

Since it's hard to look into the clustering of all 766 entities, we here show a subset of 47 animals which are subset of 6 higher order concepts ('toothed whale','rodent','deer','cat','dog','bear').
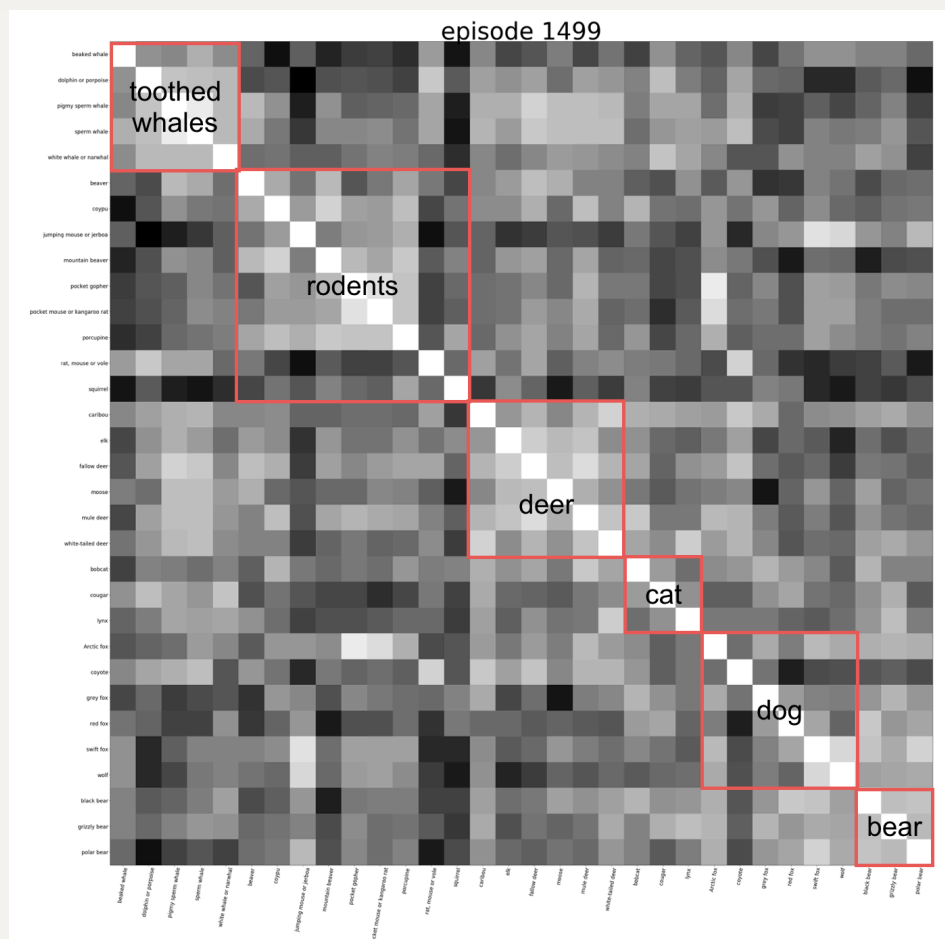
After training, certain degrees of clusters emerge. Some mammal types show more homogeneity (toothed whales, deers) than the others (rodents, dogs).

TODO: 1) subsub-concepts to show better clustering [use the linkage function; then think about how to reorder that];

3. Automated discovery

Again we replicate the method for small dataset to find new probable facts. Specifically we are interested in general concepts about higher-order categories like the ones we mentioned in last session. We set threshold to 5 and found in total 8 new relations:

'cloven-hoofed mammal-is eaten by-wolf', 'rodent-has habitat-farmlands', 'cat-chases away-opposite sex after mating', 'cat-eats-frog', 'dog-has part-short tail', 'dog-is a kind of-beaked whale', 'bear-eats-carrion', 'bear-lives in-hollow log'

Apparently most of these make sense ("dog-has part-short tail") while some are not ("dog-is a kind of-beaked whale").

4. Surprising knowledge

Another kind of discovery is to focus on surprising facts, where people can ask whether they've made some mistakes in the dataset, or there's some deeper theoretical insights. For example, people find it surprising that bat can fly yet it is a mammal, thus further questions of how bats evolve their flying skill should be asked. This is presumably a source of curiosity as well.

Here we do show an example from our dataset. We first choose a category like "dog" that includes 6 concepts, then pick the concept with least correlation with other concepts, which is "coyoto" (as can be seen from the correlation matrix in section 2). Next, we pair each relations involving coyote with the same relation but replace "coyote" to its sup-ordinate concept ("dog"), and compute the difference of prediction score from the embedding model. This brings to us the attention to some potentially surprising facts like: "coyote-makes sound-howl", "coyote-makes sound-yelp", and "coyote-is eaten by-wolf".