SingularityNET

Semantic Vision: Deep Learning + Cognitive Architectures

Alexey Potapov, Sergey Rodionov,
Maxim Peterson, Oleg Scherbakov, Innokentii Zhdanov,
Vitaly Bogdanov, Nikolai Skorobogatko,
Hugo Latapie, Enzo Fenoglio (Cisco)

AGI'18 @ Prague

Vision System for AGI: Choices

- **Level of integration into AGI**
- The amount of prior information
- Flexibility
- Generality
- Architecture
- Frameworks
- …

Vision System for AGI

- Levels of integration/specialization
  - GOFAI + classical computer vision
  - Symbolic cognitive architectures
  - Hybrid architectures
  - Deep [reinforcement] learning
  - AIXI

Vision System for AGI

- Levels of integration/specialization
  - GOFAI + classical computer vision
  - Symbolic cognitive architectures
    - Too loose integration, not enough for AGI
  - Hybrid architectures
  - Deep [reinforcement] learning
  - AIXI

Vision System for AGI

- Levels of integration/specialization
  - GOFAI + classical computer vision
  - Symbolic cognitive architectures
  - Hybrid architectures
  - Deep [reinforcement] learning
  - AIXI
    - Too general to work on a real-world high-dimensional data; specialization is required

Vision System for AGI

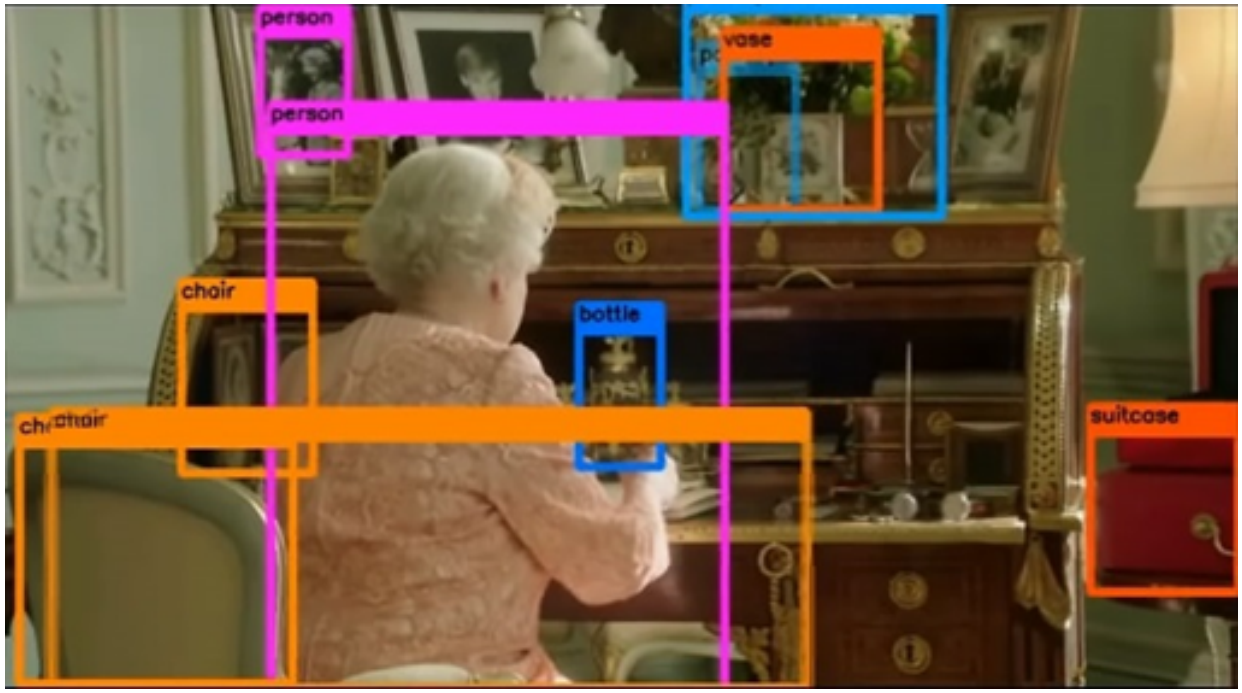
- Levels of integration/specialization
  - GOFAI + classical computer vision
  - Symbolic cognitive architectures
  - Hybrid architectures
  - Deep [reinforcement] learning
    - State-of-the-art CV models (but with some limitations)
    - Not good at cognitive tasks
  - AIXI

Vision System for AGI

- Levels of integration/specialization
  - GOFAI + classical computer vision
  - Symbolic cognitive architectures
  - **Hybrid architectures**
    - **Level of integration? Components?**
  - Deep [reinforcement] learning
  - AIXI

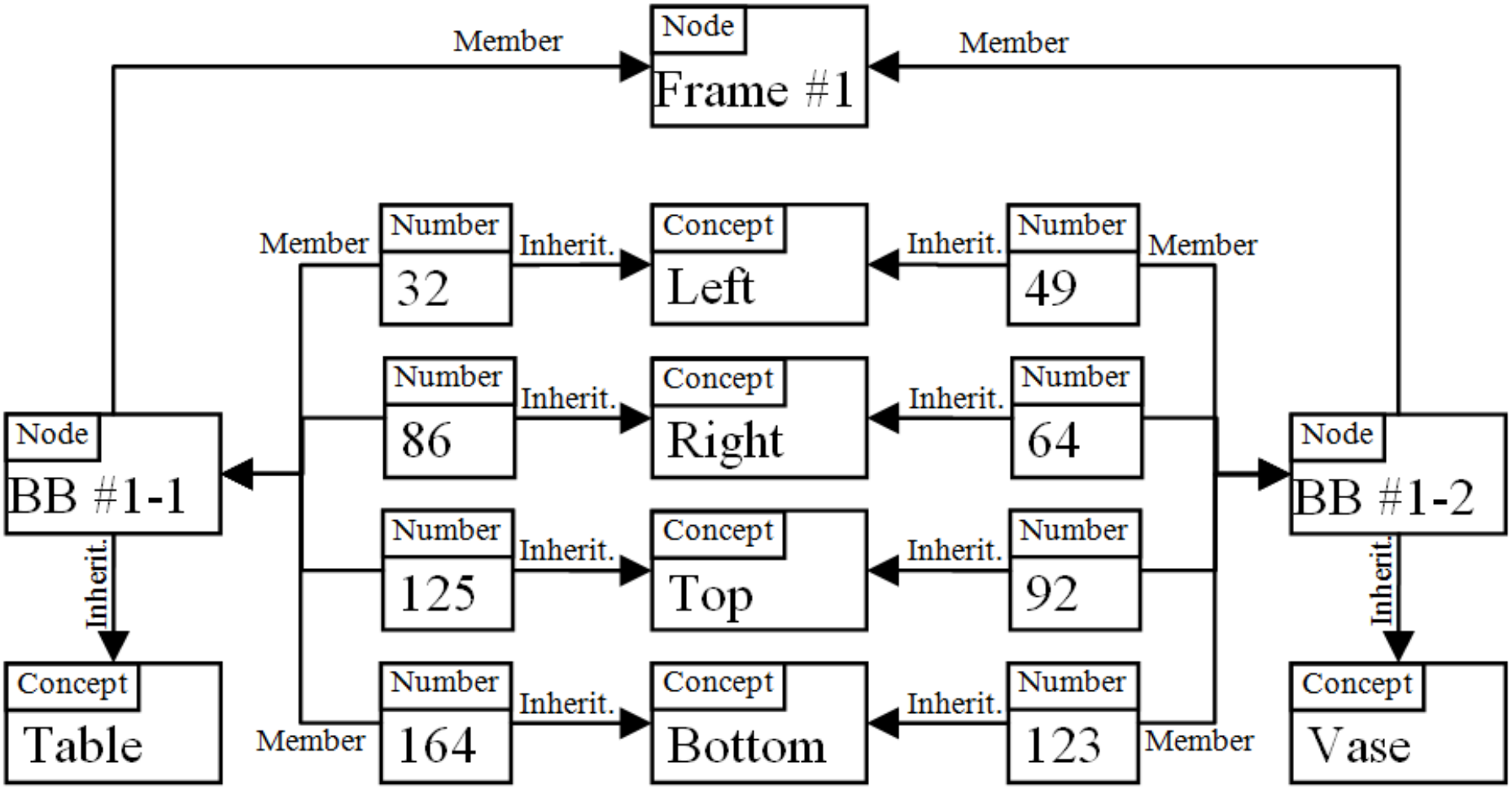
Loosest integration: semantic retrieval example

- Vision subsystem
  - Output: bounding boxes with assigned labels



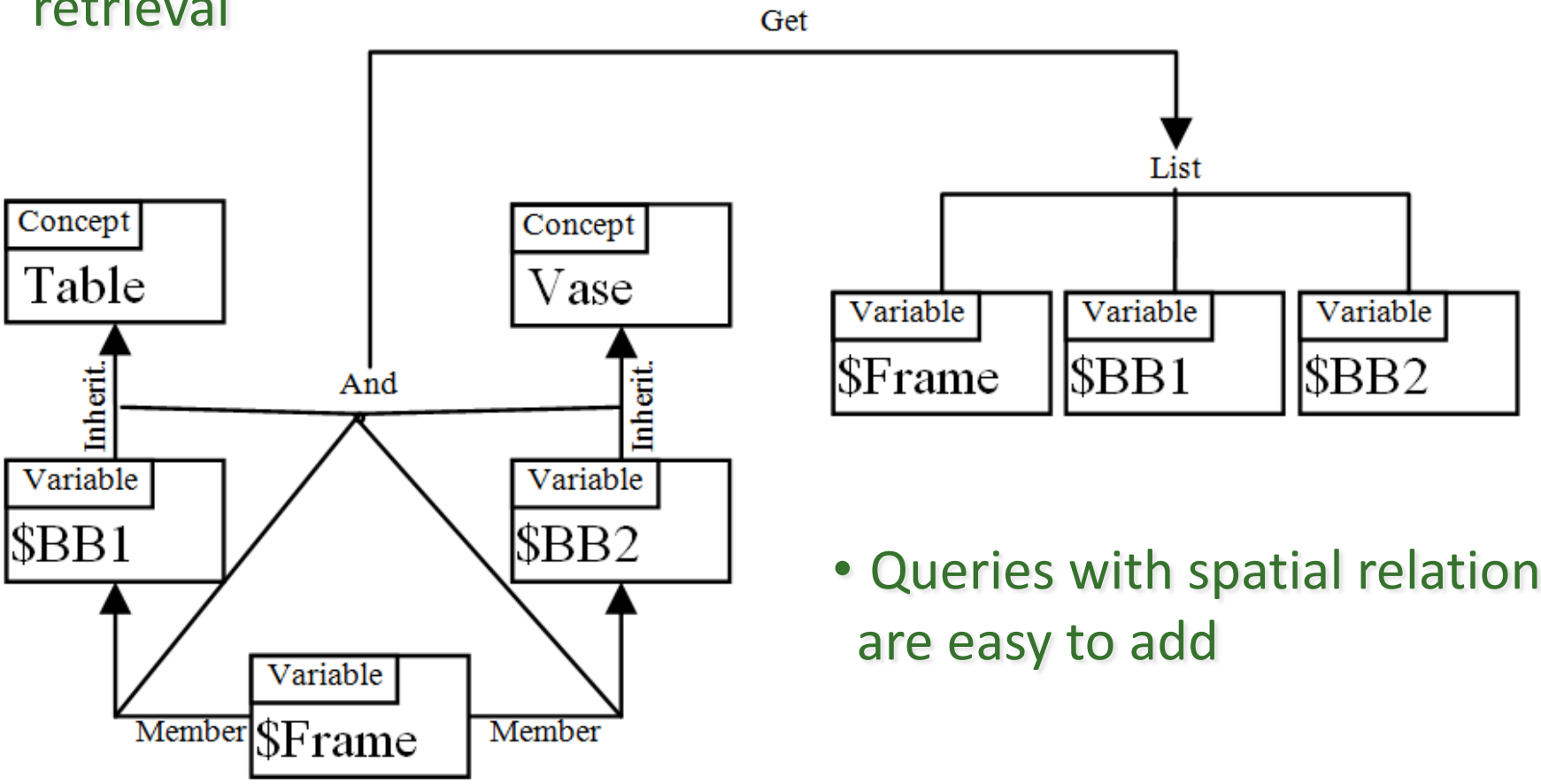
  - Inserted into: OpenCog's Atomspace

Example Representation



Queries

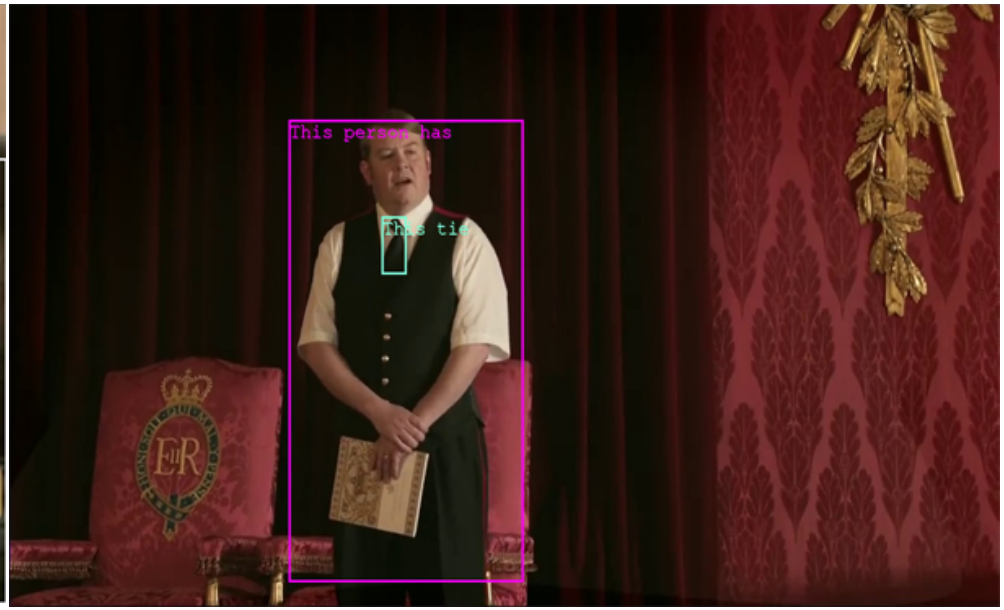
- Directly maps to OpenCog's Patter Matcher queries
- Readily provides means for the content-based image retrieval



- Queries with spatial relations are easy to add

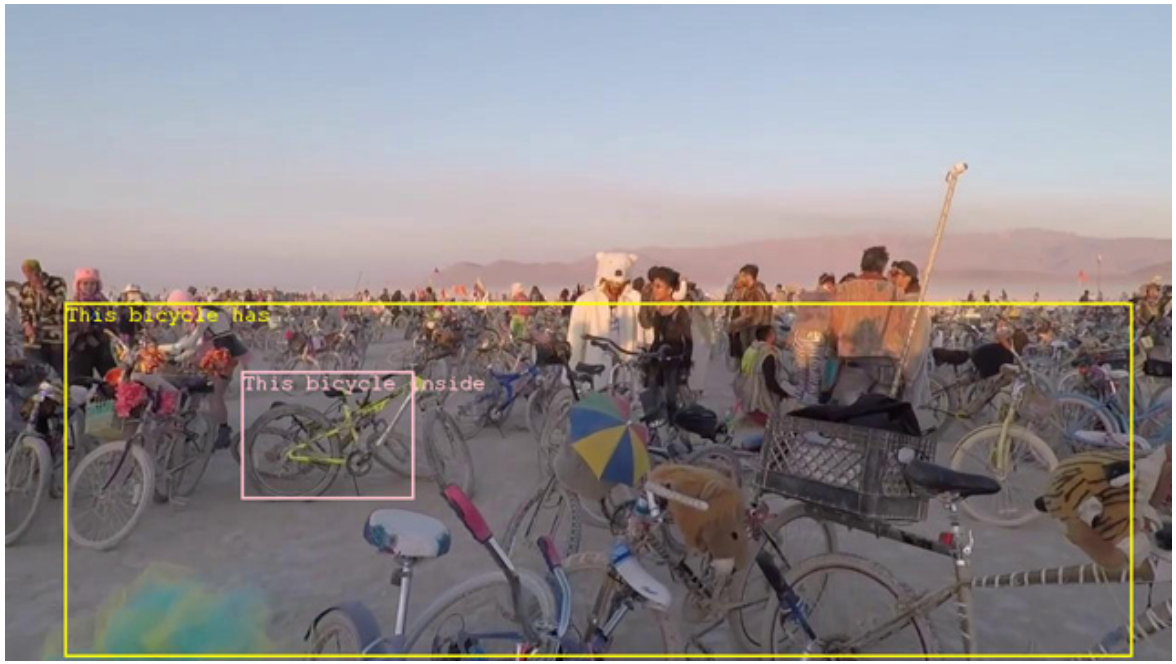
Positive examples

- A person in a car



- A person with a tier

Mistakes



- A bicycle inside a bicycle
- A person with a backpack

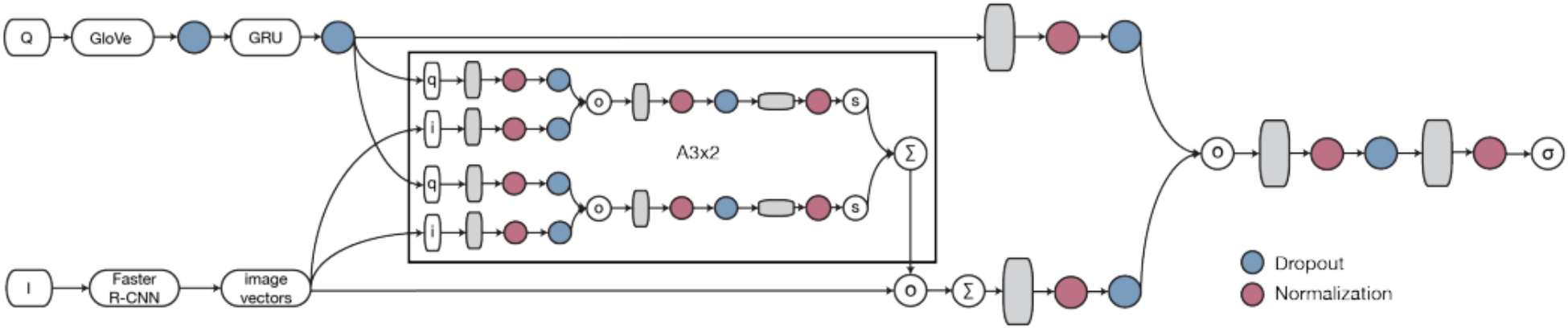
- Most problems are due to the incorrect detection and labeling of bounding boxes
- Is loose integration enough? Are vision system improvements only needed?

Visual Question Answering

- “Is the boy blond?”, “What is the child doing?”, “Is he happy?”
  - One label per object is not enough. “Boy”, “he”, “child”, “blond”, “jumping”, “happy”, etc. can be attributed to one BB, and it is not feasible to assign all sensible labels in a bottom-up way
- “Are people looking in the same direction?”
  - Labels are not enough. Each object can be described with different attributes and relations. Bottom-up extraction of all of them will result in a combinatorial explosion
- “Are the chairs similar?”
  - Symbolic descriptions are simply not enough. The cognitive system should have access to visual features

=> Tighter integration is necessary

Former state-of-the-art VQA models



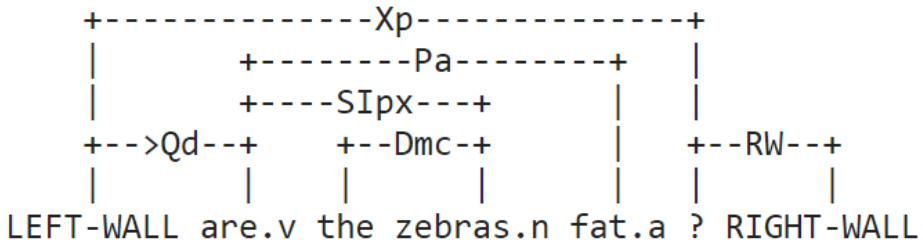
- Pre-trained word embedding
- Pre-trained object proposal and image feature extractor
- Element-wise product of question embedding and visual features
- Classifier as output

=> It's OK to use bounding boxes and highest-level visual features for now…

Singh J., Ying V., Nutkiewicz A. Attention on Attention: Architectures for Visual Question Answering (VQA). arXiv:1803.07724

Questions to Atomese

- Some questions can be directly mapped to Pattern Matcher queries
  - Are the zebras fat?
- Link grammar tree
  - (S are.v (NP the zebras.n) (ADJP fat.a) ?)



- Relex form
  - _predadj(zebra, fat)
- Pattern Matcher query (Scheme/Atomese):
  - (Satisfaction
    (TypedVariable (Variable "$X") (Type "ConceptNode"))
    (And
      (Inheritance (Variable "$X") (Concept "BoundingBox"))
      (Evaluation (GroundedPredicate "py:runNN") (List (Variable "$X") (Concept "zebra")))
      (Evaluation (GroundedPredicate "py:runNN") (List (Variable "$X") (Concept "fat")))))

Grounded Predicates

- Implementation for OpenCog
  - Technical/Conceptual issue: Atoms vs Values
  - Simplest approach
    - Each predicate (“boy”, “car”, “blond”, “jumping”, …) is grounded by its own network: $P_{word}(\mathbf{x}_{vis} | \mathbf{w}_{word})$
    - Similar to separate outputs of the classifier
    - Pattern Matcher automatically selects which predicates to evaluate
- An issue to be addressed: predicates are not independent. Can we really train DNNs to estimate e.g. P(jumping) independently of the object?

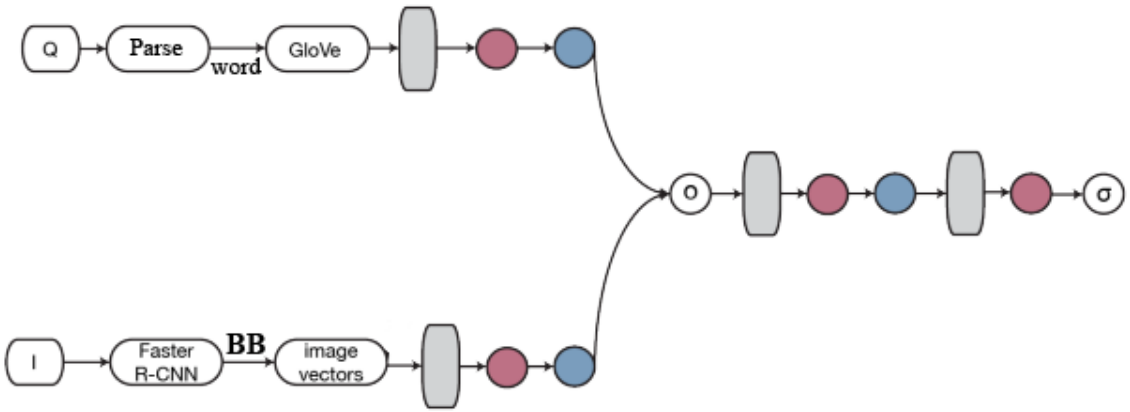
Embeddings

- Training separate predicates for words is wasteful: words can be related

⇒Utilizing word embeddings as inputs to two-argument or second-order predicates:

$$P(\mathbf{x}_{vis}, \mathbf{emb}_{word} | \mathbf{w}) \text{ or } P(\mathbf{x}_{vis} | \mathbf{w}=\text{DNN}(\mathbf{emb}_{word}))$$

- Simplest implementation can be borrowed from the DNN models:



Questions to Atomese

- Slightly more complex questions: “What color is the plane?”
  - Relex form: _det(color, _$qVar);_obj(be, plane);_subj(be, color)
  - PM query: (Bind

```
(Variable (TypedVariable (Variable"$B") (Type "ConceptNode"))
                (TypedVariable (Variable "$X") (Type "ConceptNode")))
(And
  (Inheritance (Variable "$B") (Concept "BoundingBox"))
  (Inheritance (Variable "$X") (Concept "color"))
  (Evaluation (GroundedPredicate "py:runNN") (List (Variable "$B") (Concept "plane")))
  (Evaluation (GroundedPredicate "py:runNN") (List (Variable "$B") (Variable "$X"))))
(List (Variable "$B") (Variable "$X") (Concept "plane")))
```

  - Inheritance links are necessary
    - Can be automatically extracted from questions
  - Same predicates are used both for “attention” and “classification”

Challenges

- Models for predicates involving several bounding boxes or whole image (topmost object, similar objects, counting, etc.)
- Propagating training signals for DNNs backward through symbolic inference in arbitrary case
- Learning language together with the vision subsystem:
  - Syntax to parse questions
  - Semantics in terms of PM queries and visual groundings
    - Considered questions can be directly mapped to PM queries, but how can this mapping be learned?
    - Some mappings are far from direct: “Is it raining?” → Check for clouds and umbrellas
- Extension to other tasks (text->image, image->text, semantic segmentation, SLAM, etc.)

Vision System for AGI

- What is the task?
- Discriminative models:
  - Deep Q-learning, classifiers, object detectors, etc.
    - Trained for a particular dataset/task/ reward function/… not too AGI-ish

Vision System for AGI

- What is the task?
- Discriminative models
- Generative models:
  - GANs… AIXI
    - Ultimately, reconstruct a generative model of the world

Vision System for AGI

- What is the task?
- Discriminative models
- Generative models:
  - GANs… AIXI
    - Separation of the vision task

$$P(z_t|x_t) = \frac{P(x_t|z_t)P(z_t)}{P(x_t)} = \frac{o(x_t|z_t)\int \mu(z_t|z_{t-1},a_t)P(z_{t-1}|x_{t-1})dz_{t-1}}{\int P(x_t|z_t)P(z_t)dz_t}$$

infer a description z_t of a scene using current image x_t, a generative image model o, and an environment model μ

Vision System for AGI

- What is the task?
- Discriminative models
- Generative models:
  - Still inefficient

=> Although the task of vision consists in inferring a latent description within the trainable generative model, a solution of this problem requires the construction of a system of discriminative models, both general purpose and specialized, with possible interaction with the generative model

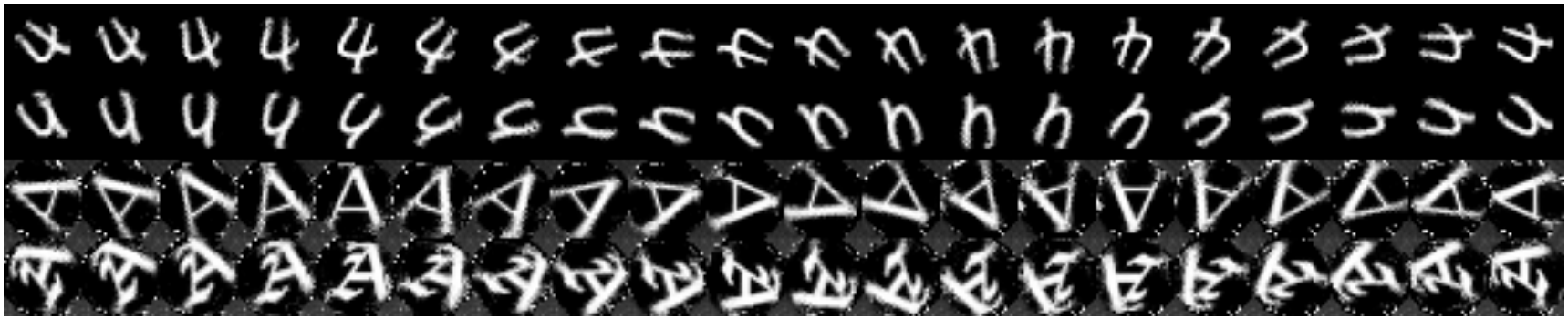
Frameworks

- Handcrafted CV systems
  - obviously not enough (both for narrow and general AI)
- Machine Learning
  - Deep Learning
    - Is it enough?
  - Probabilistic programming
    - Still lack computational efficiency

DNNs: What can be learned?

- Both traditional DNNs and CapsNets fail to learn invariants (for recognition) or to generalize transformations (for generation)
- They also fail to learn semantically meaningful visual concepts in complex domains without supervision

DNNs: What can be learned?



- E.g., HyperNets can learn to transform or normalize images independent from its content

- But should we require this from the vision subsystem, or is this a problem for AGI in total?

Questions

- Do we require the capabilities to learn invariants and extract hierarchical relations from the discriminative models?
- Should generative models be normally involved in image analysis?
- Should the generative subsystem infer such latent variables, which are not estimated by the discriminative subsystem?
- How strong priors should be?
- Should the generative and discriminative models be aligned on all levels? Should we use traditional neural networks for discriminative models? How should we extend existing formalisms for generative models? What are acceptable architectures for vision tasks beyond object recognition?

SingularityNET

Thank you for attention!

Contact: alexey@singularitynet.io