Teadmiste formaliseerimine

aka

Knowledge Representation
(and Reasoning)

KR & R

Intro

Course page

http://lambda.ee/wiki/Teadmiste_formaliseerimine

Course theme

SQL & conventional software <----> natural language

- Structured facts: SQL
- Unstructured knowledge: natural language
- *Focus of the course: semi-structured knowledge*

Course theme

SQL & conventional software <----> natural language

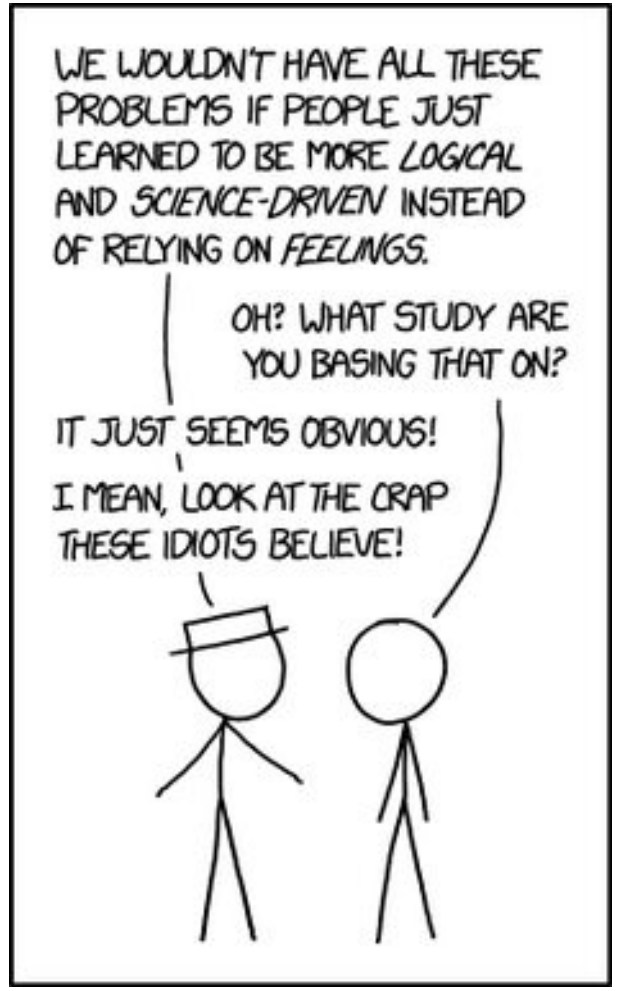
- Structured facts: SQL
- Unstructured knowledge: natural language
- *Focus of the course: semi-structured knowledge*

  Capturing meaning in natural language sentences:

  understanding what is said, answering questions, explaining answers, calculating plans of action.

Question answering and inference using NL

- Strongly related to commonsense reasoning.
- Long history, little success with symbolic methods. CYC!?
- Large Language Model (LLM) explosion: success with issues
  - Explainability
  - Hallucinations
  - Possible, but hard to connect with tools like databases etc
  - Possible, but hard to do search: deep inference, planning
  - Controllability: how to add a lot of rules?
  - etc



Can we “fix” end-to-end learning?

Nobody knows so far.

A lot of people put their hope on hybrid systems:

> Combine learning with search, rule-based reasoning, arithmetic, SQL etc.

Main steps of the course

SQL and logic

SQL and triples / RDF / JSON-LD

Simple rules like in RDFa

Large semi-structured knowledge bases

Machine learning in NLP: vector representation of words

Machine learning in NLP: large language models (BERT, GPT)

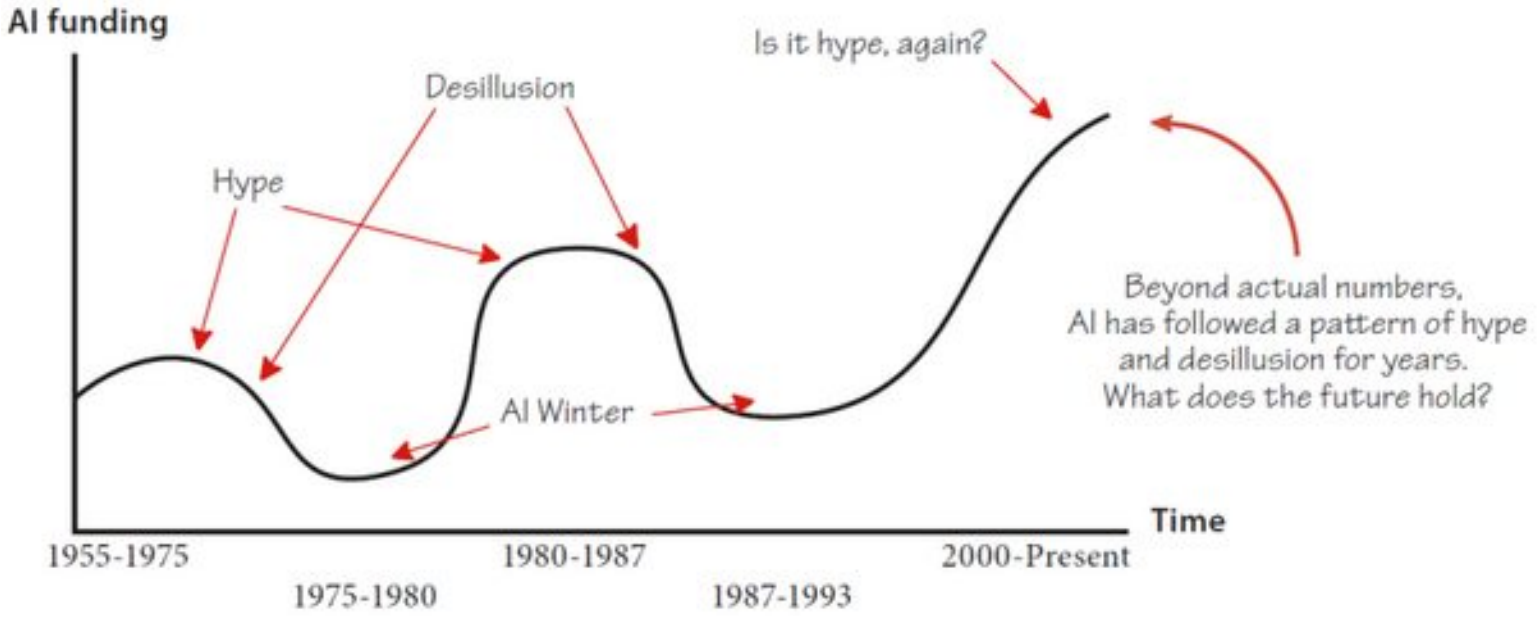
Reasoning with uncertain knowledge / uncertain rules

Semantic parsing

Hybrid question answering: machine learning + rule-based reasoning

What is pure classical KR & R?

- A classic old-school A.I. subfield *for actual human-like thinking*
- Used to set lofty goals extremely hard to achieve (and did not)
- Famous A.I. winters:



Approach to thinking in KR&R

Symbolic A.I:

- Write down (human?) knowledge using rules and facts.
- Answer questions and solve problems by deriving new knowledge using rules.

Next, small examples using an automated reasoner for first order logic: run these yourself in http://logictools.org

```
spacity(kuressaare).

railroad(tallinn,tapa).
railroad(tapa,tartu).
highway(tallinn,virtsu).
sealane(virtsu,kuivastu).
highway(kuressaare,kuivastu).
railroad(X,Y) => railroad(Y,X).
highway(X,Y) => highway(Y,X).
sealane(X,Y) => sealane(Y,X).
railroad(X,Y) => easytravel(X,Y,use(train,X,Y)).
highway(X,Y) => easytravel(X,Y,use(bus,X,Y)).
sealane(X,Y) => easytravel(X,Y,use(ship,X,Y)).
easytravel(X,Y,P1) & easytravel(Y,Z,P2) => easytravel(X,Z,combine(P1,P2)).

easytravel(tartu,X,Y) & spacity(X) => $ans(X,Y).
```

- Replace *spacity(kuressaare)* with

city(kuressaare).

has(kuressaare,coast).

hasnrof(kuressaare,spa)=4.

population(kuressaare)=10000.

$less(population(X),20000) => smallcity(X).

(city(X) & has(X,coast) & $less(3,hasnrof(X,spa)) & smallcity(X)) => spacity(X).

…

Note: it can be pretty hard to find solutions

- Finding a solution is semidecidable: no guarantee that we can show that a solution does *not* exist.

- Even small problems can be hard. A solvable tiny example from group theory, ca 57 steps in a proof (try it out on logictools.org):

multiply(A,inverse(multiply(B,multiply(C,multiply(multiply(inverse(C),inverse(multiply(D,B))),A))))) = D.

multiply(inverse(a1),a1) != multiply(inverse(b1),b1).

A narrow subfield of KR &R: automated reasoning

- AKA “Automated Theorem Proving”
- Focus on developing algorithms and systems for automatically solving problems written in logic.
- Yearly competitions http://www.tptp.org/CASC/ and the main conference CADE http://www.cadeinc.org/conferences
- Focus more on mathematical kinds of problems, not so much common sense or very large knowledge bases.
- In other words, top systems work best with relatively few rules and deep complex proofs.
- Not too many high-performance systems (like, my system Gandalf was world leading at the end of 20th century for many years, and the current GKC system is one of the top systems nowadays)

What happens with this approach?

- Small toy problems are easy to present and solve.
- Presenting and solving large complex problems for common-sense reasoning has turned out to be extremely hard.

Why is symbolic reasoning with NL hard?

- Semantic parsing, i.e. translating natural language to logic, is extremely hard due to the highly complex and exception-rich nature of natural language.
- Existing knowledge bases of "common sense" do not cover a critical mass of the basic understanding of the world even a small child possesses.
- Classical first order reasoning itself cannot cope with contradictory knowledge items, probabilistic or uncertain information and exceptions to rules.
- Finding logic-based proofs often requires long derivations and the huge knowledge base causes a quick combinatorial explosion of the search space.

Hypothesis: ML and LLMs can alleviate all the main issues above

Famous case study: CYC

Wikipedia

- **Cyc** is a long-term artificial intelligence project that aims to assemble a comprehensive ontology and knowledge base that spans the basic concepts and rules about how the world works.
- Douglas Lenat began the project in July 1984 … since January 1995, has been under active development by the **Cycorp** company.
- The Cyc project has been described as "one of the most controversial endeavors of the artificial intelligence history". Catherine Havasi, CEO of Luminoso, says that Cyc is the predecessor project to IBM's Watson. Machine-learning scientist Pedro Domingos refers to the project as a "catastrophic failure“.

A *mostly* failed limited approach: Semantic Web

- Idea: make it easy to publish, harvest, integrate and use structured knowledge on the web.
- Proposals worked out: RDF data model and OWL rule-like language

- RDF main idea: represent all data in a single three-column table of <object id> <property> <value> triplets like this:

client_1	name	„John Brown“
client_1	balance	200

What happened with the Semantic Web?

Negative:

- Mostly academic research leading to over-complicating even simple things.
- People were not really keen to publish data or rules on the web.

Positive:

- Attempt to make it a bit simpler in a closely related area: linked data
- JSON-LD language giving RDF interpretation to JSON
- http://schema.org repository of suggested type/property names

Neurosymbolic / hybrid systems

- A much smaller research area than end-to-end learning
- Most of the research investigates ways to use rules while learning
- What we are working on:
  - a symbolic pipeline for commonsense reasoning with NL, geared towards experimenting with ML and LLM augmentation
  - learning to reason in long exact chains and finding proofs
  - improving image understanding via building a textual chain of information

Research area of neurosymbolic A.I.:

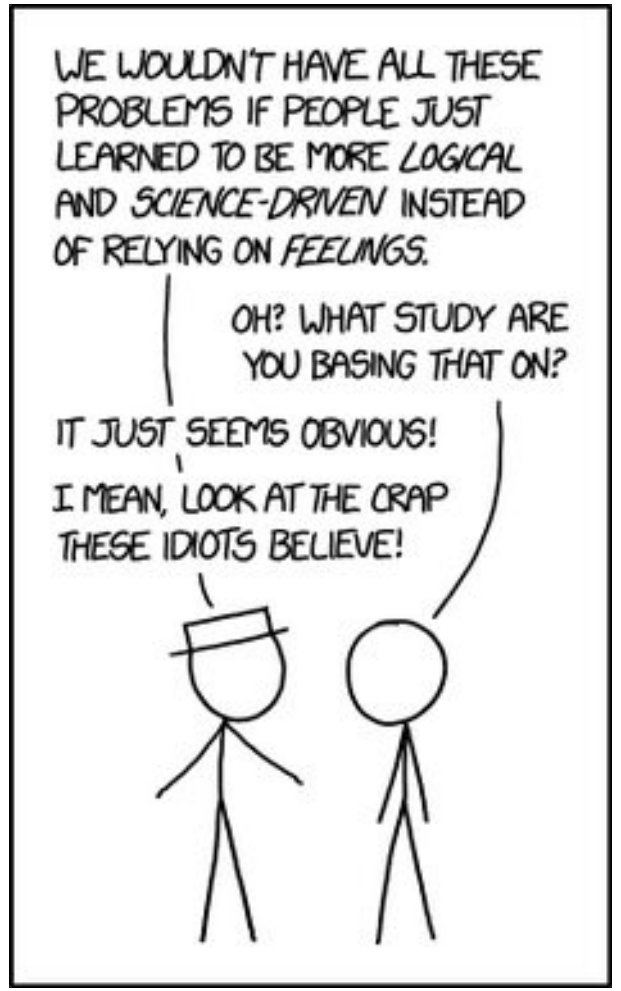
Find ways to combine M.L. with symbolic components to be able to solve different really hard tasks.

Some approaches:

- Use M.L. to heuristically guide proof search for symbolic reasoning
- Build LLMs capable of autoformalization and then use symbolic reasoning
- Let LLMs learn on symbolic knowledge in addition to free text
- Build new architectures for M.L. capable of learning external software tool use
- Build new architectures for M.L. capable of performing search / planning
- Build symbolic rules already into a neural switch level of M.L. systems

Our umbrella project: autoformalisation using extended first order logic

Autoformalisation: convert tasks presented in natural language to a formal representation usable by specialized tools (planners, data aggregators, math, logical reasoners etc).



Can we pre-build some necessary structure?

Like making the system internally know things like:

- There are objects, and they are mostly persistant.
- Objects are in space and time and follow some basic physics.
- Objects have properties.
- Objects have relations between each other.
- Relations have some basic logical properties.
- Some things are dangerous, some are desirable

…

Hybrid systems: ML + rules

Three approaches

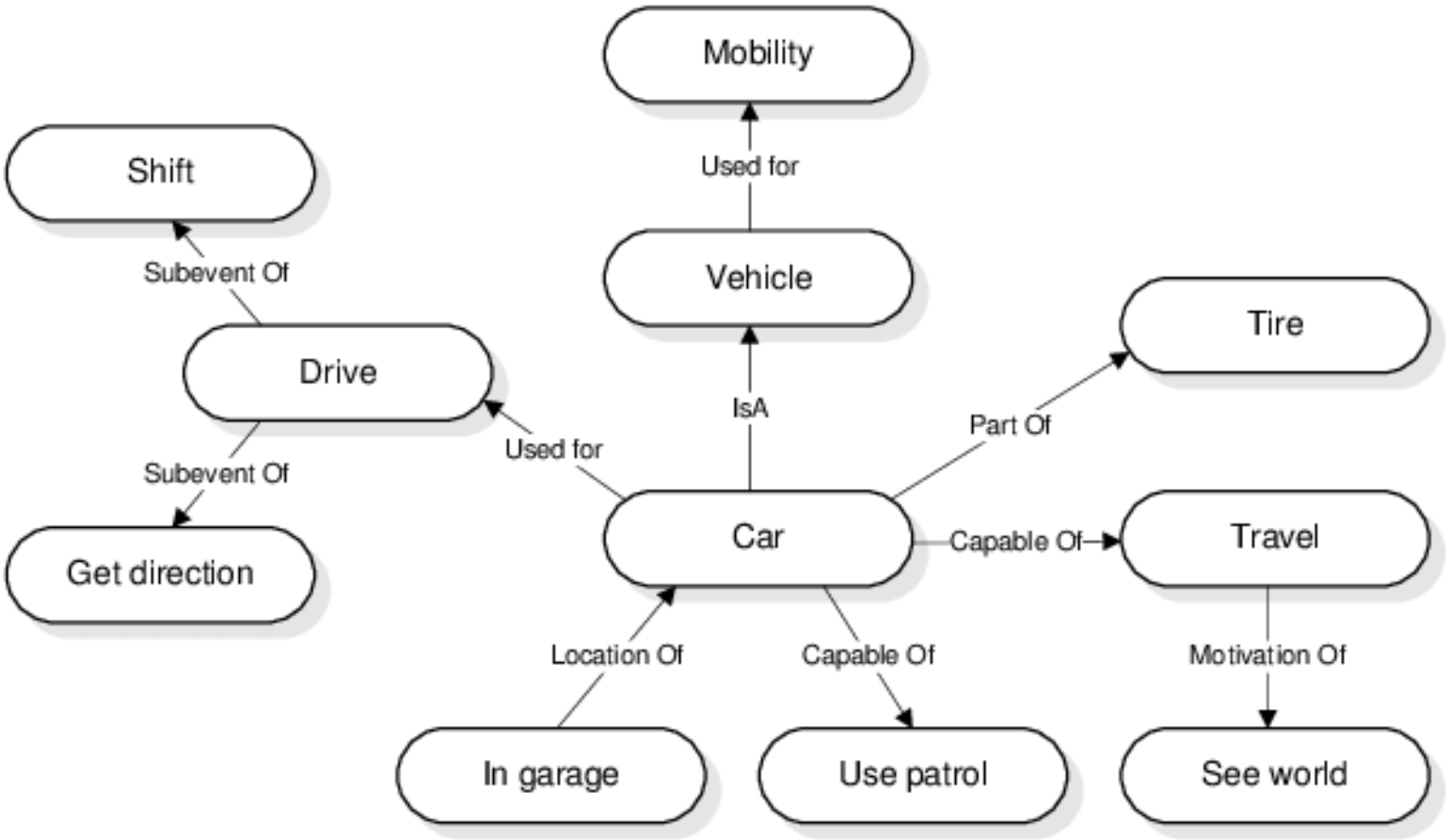
- Inject rules into a machine learning model (extremely active field)
- Use LLMs for autoformalization and then use special tools (active)
- Add machine learning to a rule-based system (only a few groups)

One big issue: good commonsense KBs?

Some of the most well-known:

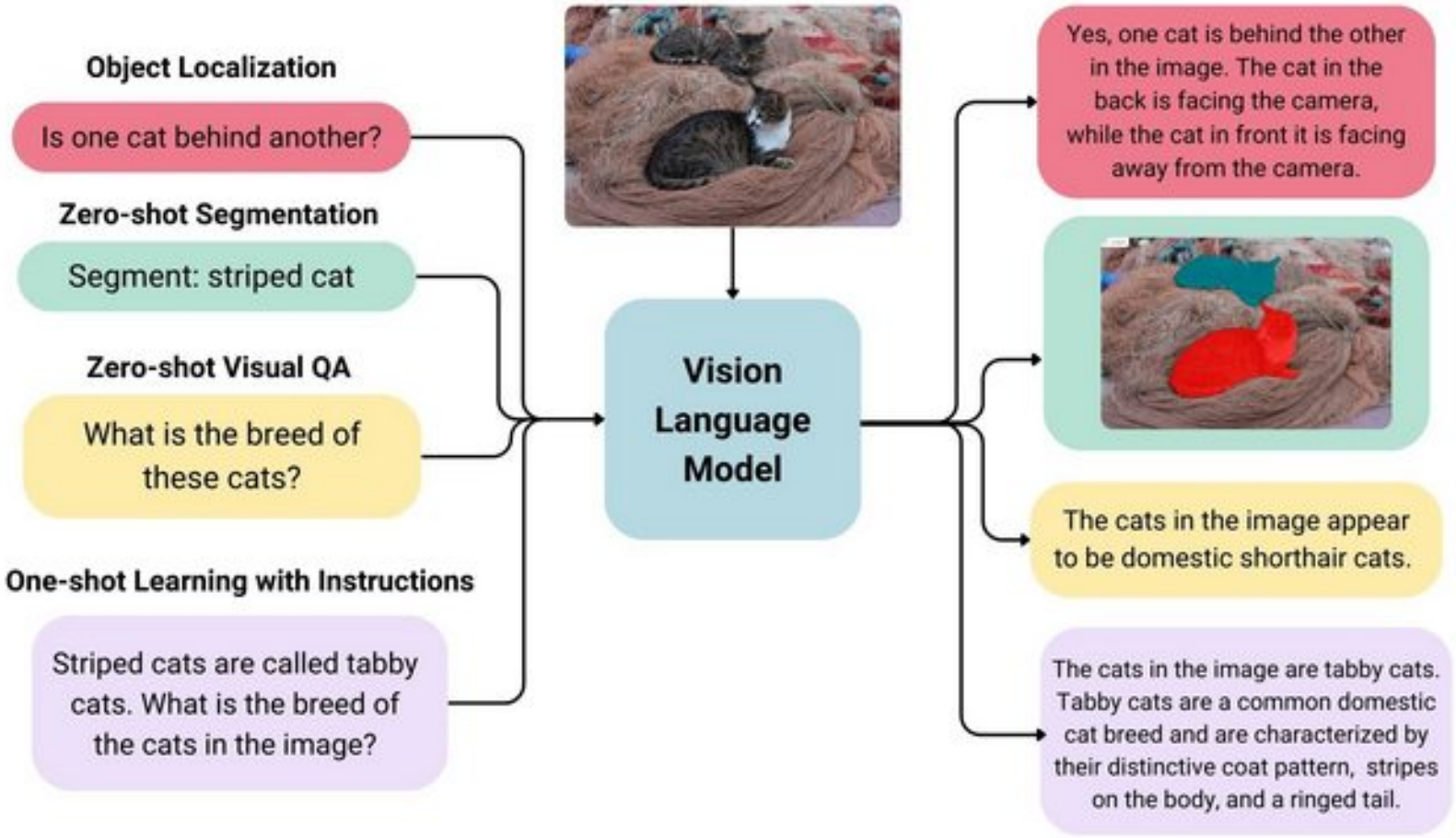
- wordnet
- Dbpedia
- wikidata
- yago
- babelnet
- conceptnet
- atomic
- nell
- framenet
- cyc
- sumo
- TPTP
- schema.org

Typical KG contents: simple triplets like



Vision language models

- Learn simultaneously from images and texts.
- Tackle tasks from visual question answering

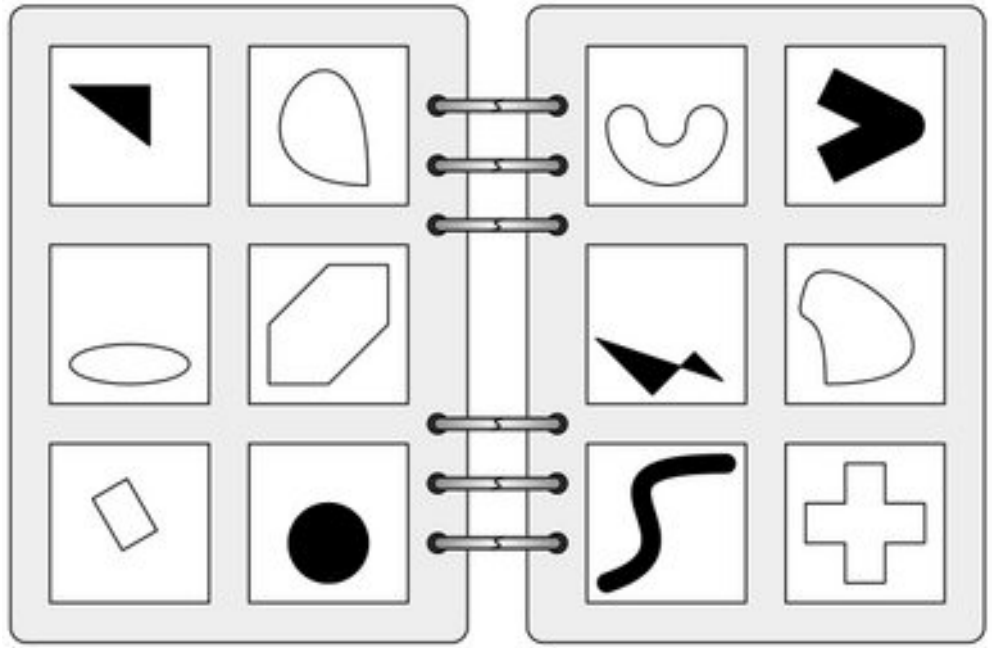


Visual reasoning and Bongard problems

Joint work by Mohit Vaishnav, Tanel Tammet, Enrico Vompa

Illustration of a Bongard OpenWorld dataset showing three *positive* and *negative* samples along with a query image. Distinguishing rule in this example is "A group photo at a wedding reception". Based on the rule, query image belongs to *negative* category.

Natural and synthetic Bongard problems



Idea: compare three different ways to find an answer using VLMs

Input: 6 pos, 6 neg images, 1 test image

Textual rule and description

Structured representation

Predicted category: pos or neg

Arrows represent corresponding questions to LLM