High-Precision Semantics Extraction in STEM

Jan Frederik Schaefer
Supervisor: Michael Kohlhase

FAU Erlangen-Nürnberg

CICM 2020 — Doctoral Program
remotely from Erlangen, Germany

My Situation

- Finishing up my master studies
- Did already some research
  - Symbolic natural language semantics
  - Controlled natural languages for mathematics
- Going to start my PhD soon
  - Supervisor: Michael Kohlhase *kwarc group*
  - Tentative topic: *high-precision semantics extraction in STEM*
  - Topic still very flexible

  → **Any feedback appreciated!**

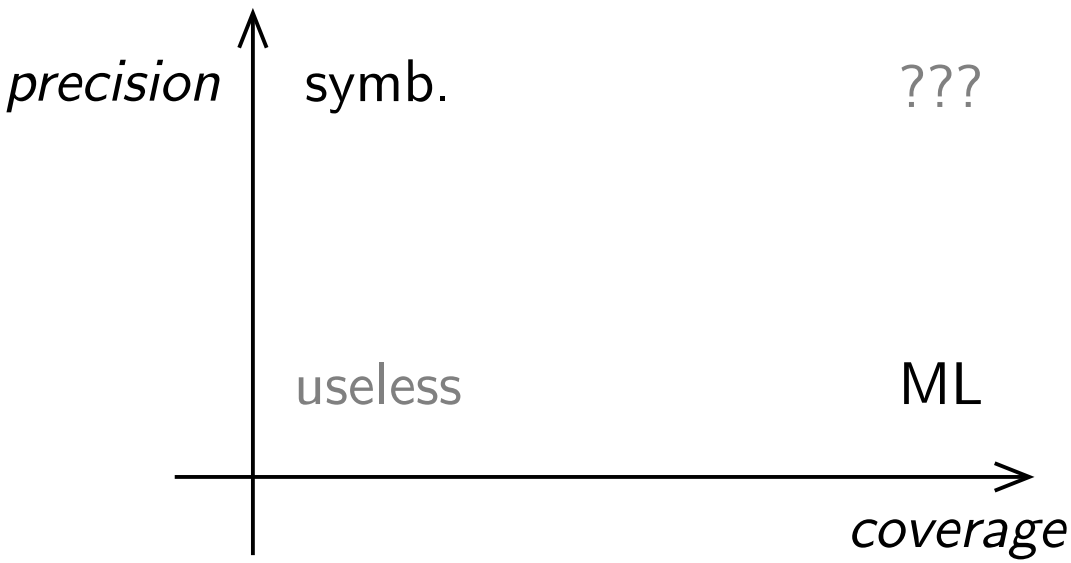
Motivation

- We have large corpora of STEM knowledge *e.g. arxiv*
- Computers can make it more accessible:
  - Unit conversion
  - Applicable theorem search
  - Screen readers
  - ...
- Such services require/benefit from semantic information
- Authors often don't provide much *semantic TEX macros*

→ Semantics extraction

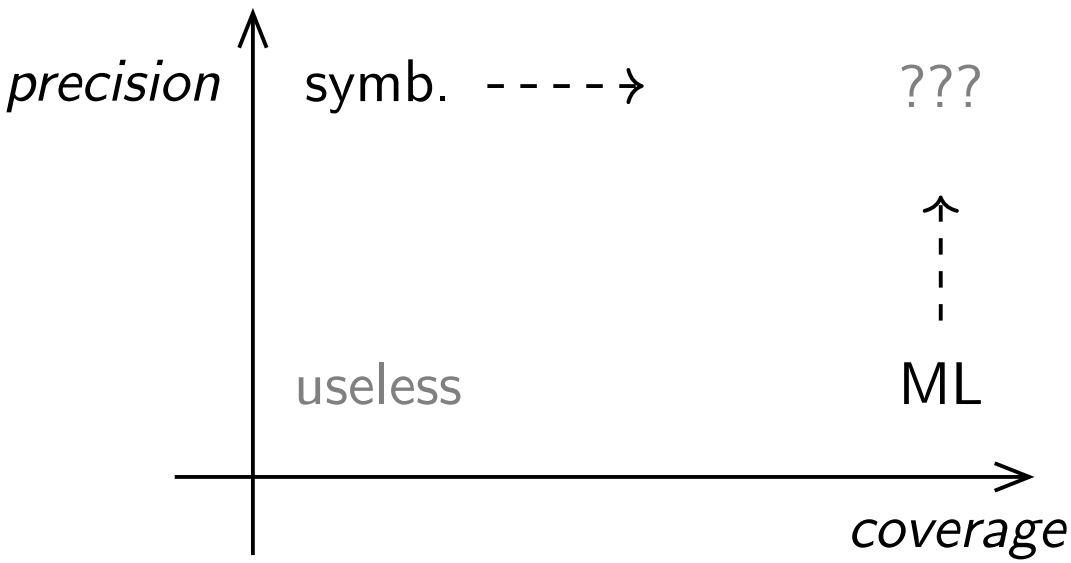
Approaches to Semantics Extraction

- Machine learning–based
  - Training data?
  - Low precision
- Symbolic
  - Low coverage



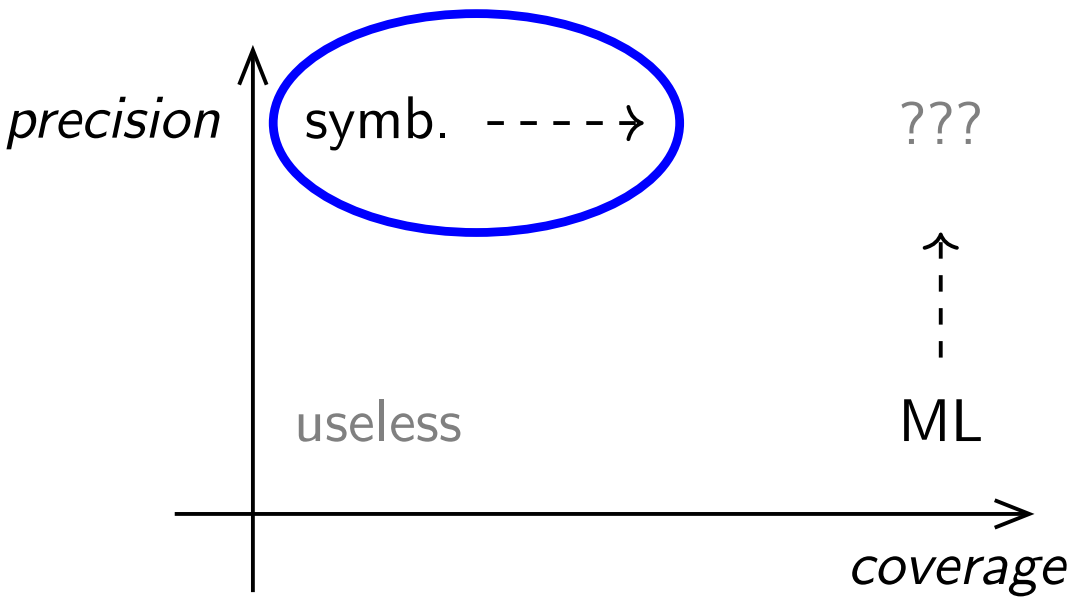
Approaches to Semantics Extraction

- Machine learning–based
  - Training data?
  - Low precision
- Symbolic
  - Low coverage



Approaches to Semantics Extraction

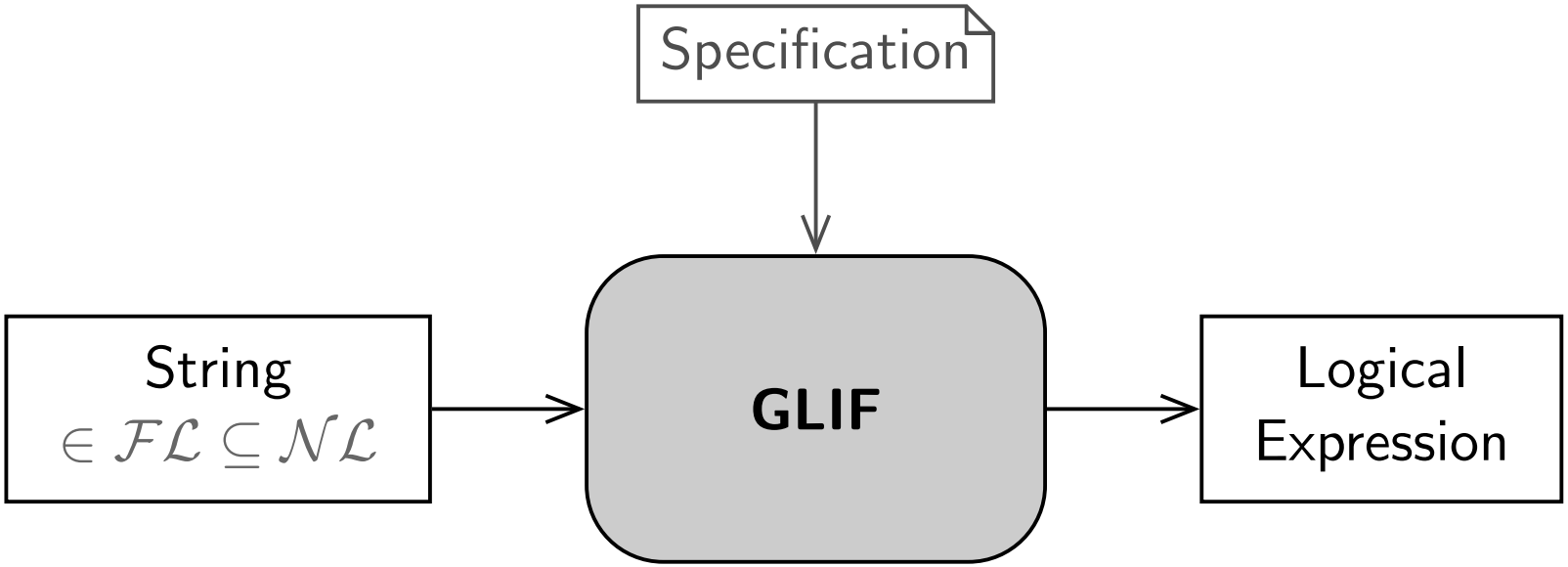
- Machine learning–based
  - Training data?
  - Low precision
- Symbolic
  - Low coverage



Plan

- Symbolic approaches offer high precision
- Often, high precision is more important than coverage
  "*each has the mass* $\frac{1}{2}m$" vs "*each has the mass* $1.64ft$"
- We already have a tool: GLIF
- Use statistical methods to increase coverage

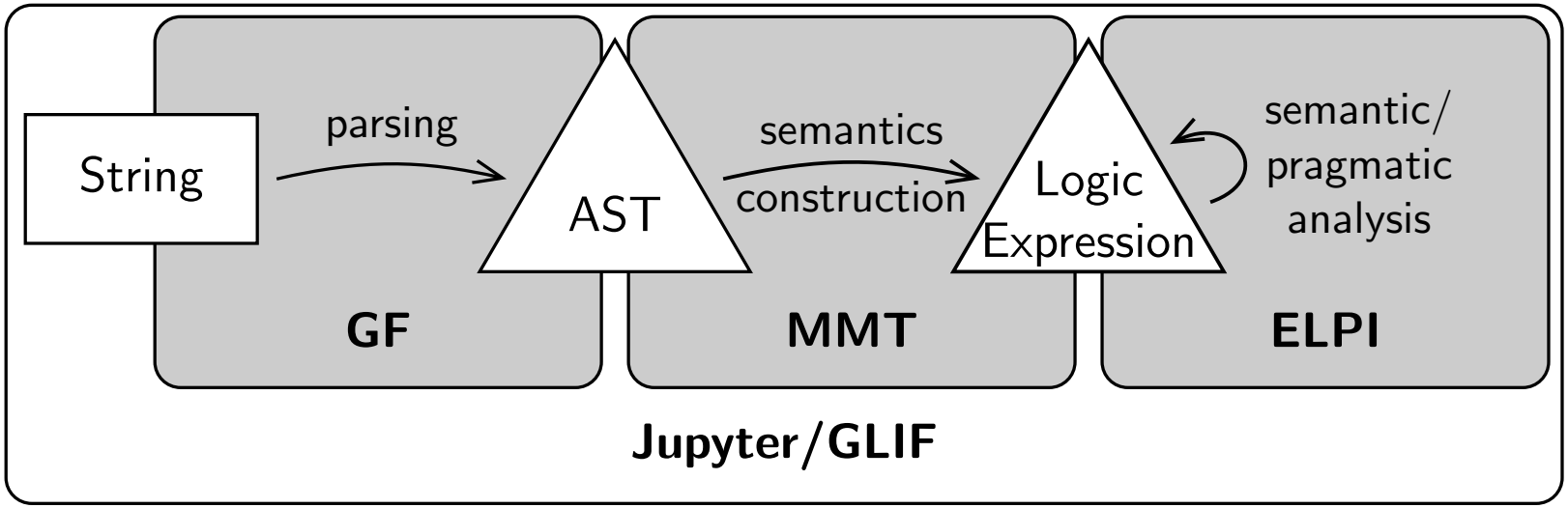
GLIF (Grammatical Logical Inference Framework)



Use cases:

- Designing controlled natural languages
- Prototyping approaches to natural-language semantics

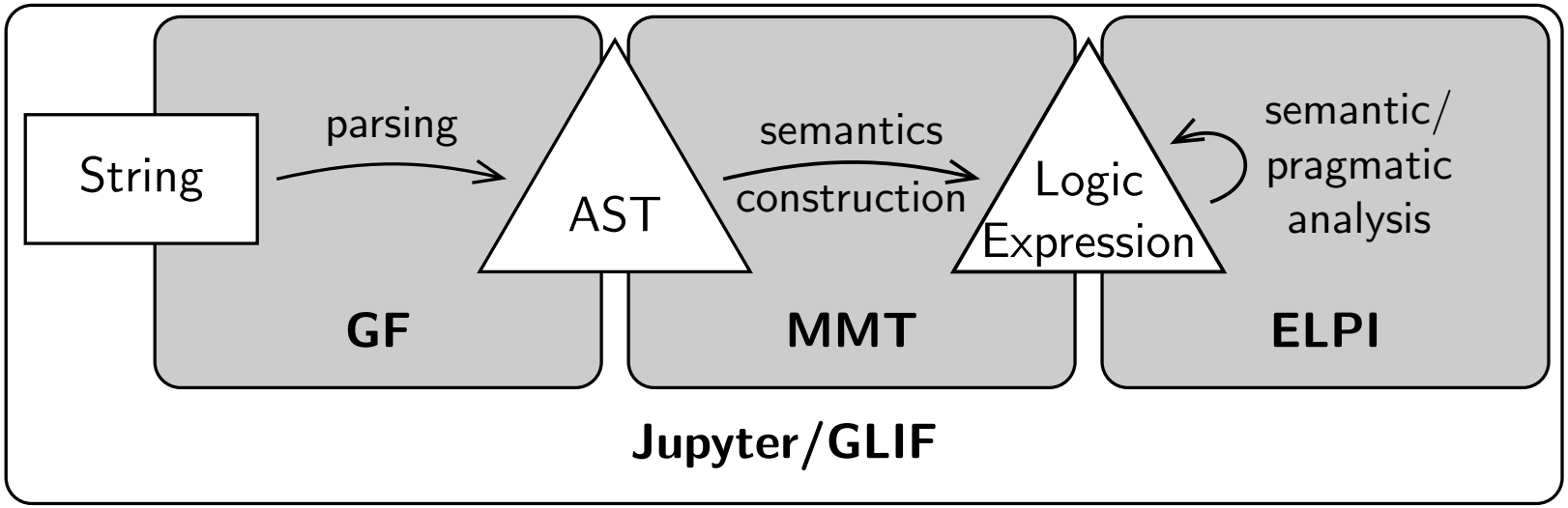
GLIF (Grammatical Logical Inference Framework)



Idea: Combine existing frameworks

- GF (Grammatical Framework): $\mathcal{NL}$ grammars
- MMT: Logic, knowledge representation
- ELPI ($\supseteq$ λProlog): Inference
- Jupyter: Intuitive UI

GLIF (Grammatical Logical Inference Framework)

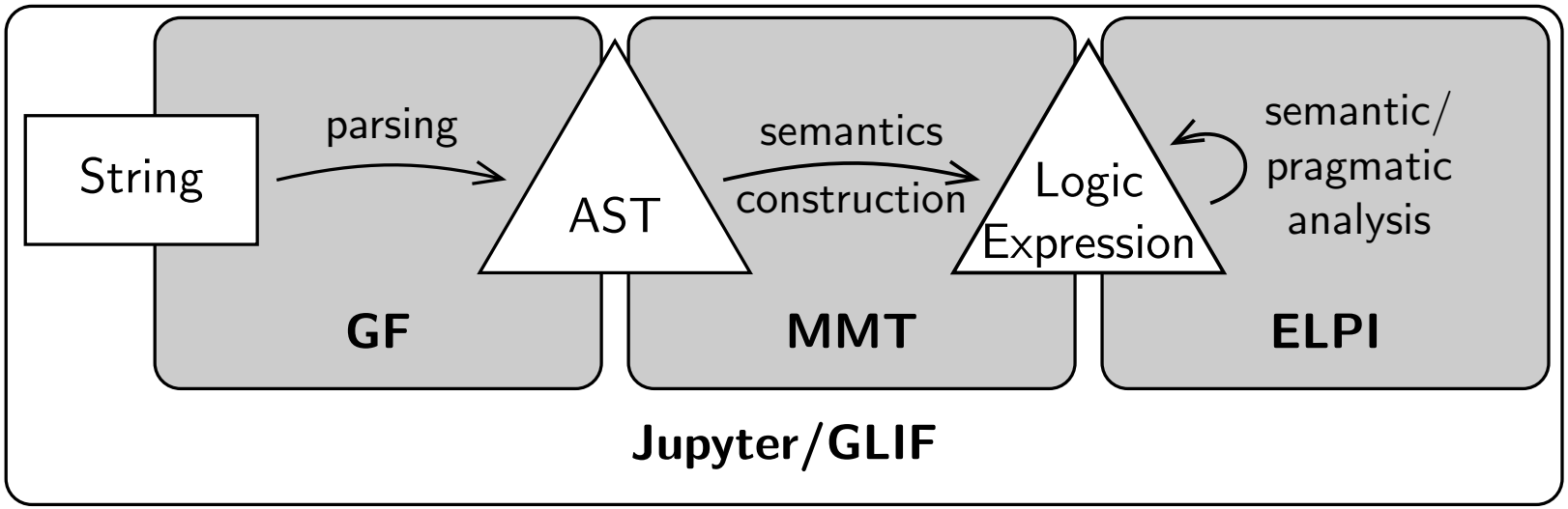


"*. . . has a mass of 2m*"

$\text{AST}_1 \longrightarrow \lambda x.\text{mass}(x, \text{quant}(2, \textbf{meters}))$

$\text{AST}_2 \longrightarrow \lambda x.\text{mass}(x, \text{mul}(2, \textbf{mVar}))$

GLIF (Grammatical Logical Inference Framework)



"*. . . has a mass of 2m*"

$\text{AST}_1 \longrightarrow \lambda x.\text{mass}(x, \text{quant}(2, \textbf{meters}))$

$\text{AST}_2 \longrightarrow \lambda x.\text{mass}(x, \text{mul}(2, \textbf{mVar}))$

Lexicon Extension

"Therefore, A is clopen."

Different options:

1. Use a dynamic parser *DynGenPar*
2. Generate lexicon automatically:
   `clopen_Adj : Adjective = "clopen"`
   `clopen :` $\iota \to o$
3. Replace lexicon entries with tokens:
   "Therefore, A is `ADJ-1`*."*

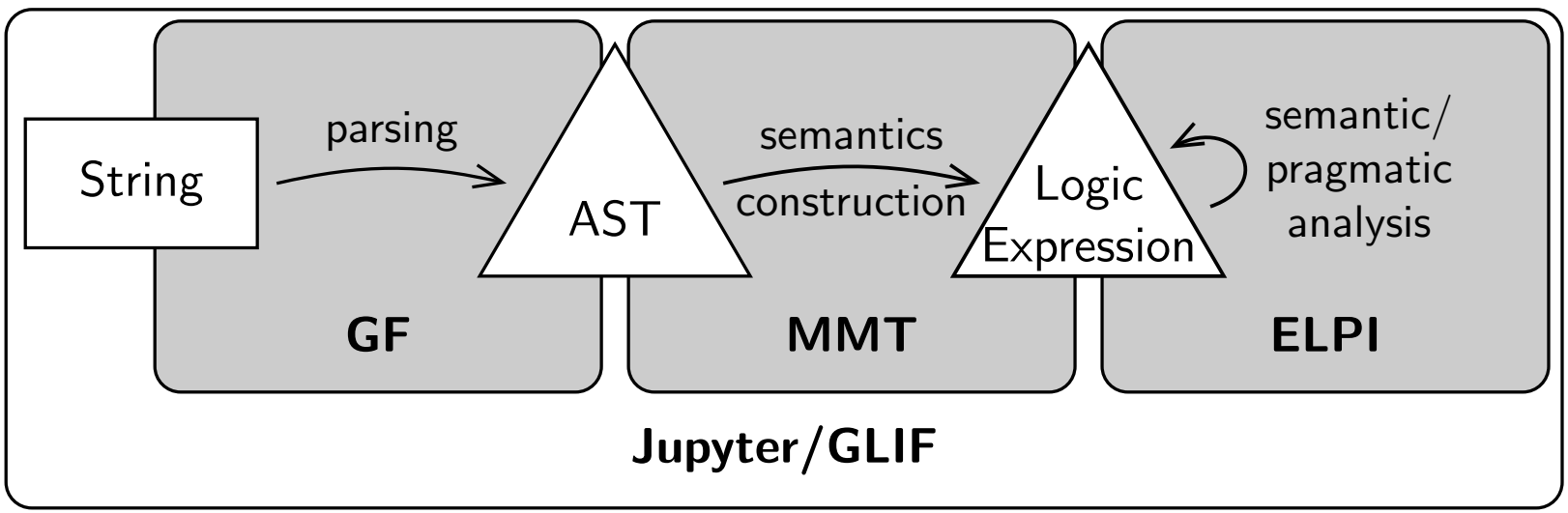
Blanking out Unparsable Parts

This may be impossible:
"Let X denote a data set where all entries $x \in X$ are normalized as described above."

This would still be useful:
"Let X denote a data set where `SUB-CLAUSE`*."*

Semantic Representation?

- Open question
- VIP, Naproche use DRT
- DRT not really supported by GLIF/MMT yet



Late Disambiguation

1. Syntactic disambiguation:

   "*2*m" → *unit?*

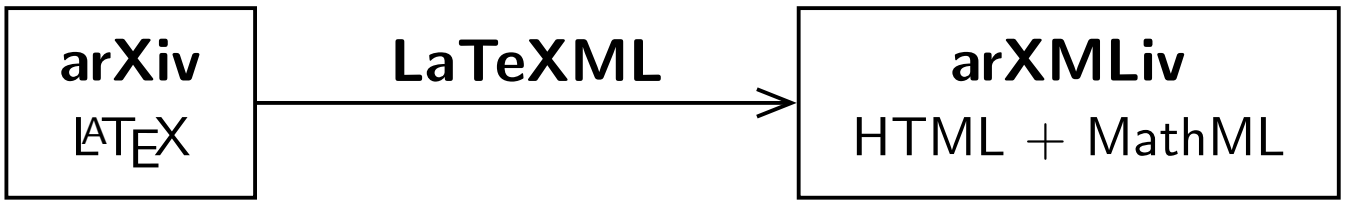
2. Semantic disambiguation:

   "*a mass of 2*m" → *not a unit*

3. Later semantic disambiguation:

   "*it has a length of 2*m*, where m is the length of a module*" → *not a unit*

Data Set



- $> 10^6$ documents
- aims to preserve any semantic information from LaTeX sources
- have some experience with processing **arXMLiv** documents

`https://sigmathling.kwarc.info/`

Work Plan

1. Prototype GLIF pipeline
   - Target: variable declarations and uses
   - Use generated lexicon
2. Prototype pipeline for corpus work
   - Load document
   - Enter pre-processed sentences into GLIF pipeline
   - Export results
3. Introduce blanking out
4. Scaling
   - Larger grammar
   - More semantic phenomena
5. Build example semantic services
6. Can we replace more with ML?
   Can the results be used as training data?

Discussion

- Is it desirable? *I think so*
- Could this work?
- Other ideas?
- Anything else?

I haven't started yet → any feedback is welcome!