Helping Humans Align Efficiently

Jan Frederik Schaefer, Michael Kohlhase

FAU Erlangen-Nürnberg

Workshop on Math Datasets: alignments and comparisons (ALIGN2025)
Brasilia, Brazil
October 10, 2025

Introduction

- Alignments support FAIR math data *Findable, Accessible, Interoperable, Reusable*

Introduction

- Alignments support FAIR math data *Findable, Accessible, Interoperable, Reusable*
- Example alignments:

  `omp2:E2119` (“Natural number”)

  `wd:Q21199` (natural numbers, possibly including 0)

Introduction

- Alignments support FAIR math data *Findable, Accessible, Interoperable, Reusable*
- Example alignments:

`omp2:E2119` (“Natural number”)
`wd:Q21199` (natural numbers, possibly including 0)
`wd:Q28920044` (positive integers, i.e. natural numbers excluding 0)
`wd:Q28920052` (non-negative integers, i.e. natural numbers including 0)

Introduction

- Alignments support FAIR math data *Findable, Accessible, Interoperable, Reusable*
- Example alignments:

  `omp2:E2119` (“Natural number”)

  `wd:Q21199` (natural numbers, possibly including 0)

  `wd:Q28920044` (positive integers, i.e. natural numbers excluding 0)

  `wd:Q28920052` (non-negative integers, i.e. natural numbers including 0)

  “Let n be a natural number.”

Introduction

- Alignments support FAIR math data *Findable, Accessible, Interoperable, Reusable*
- Example alignments:

  `omp2:E2119` ("Natural number")

  `wd:Q21199` (natural numbers, possibly including 0)

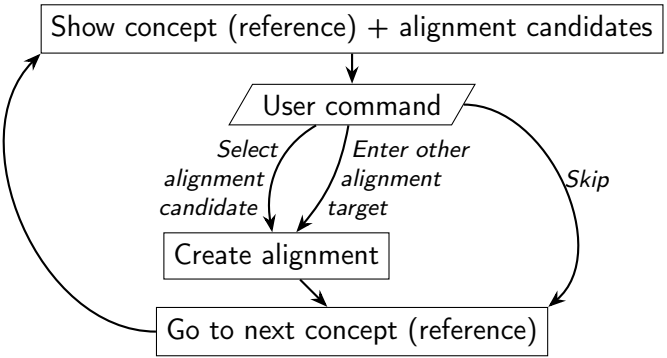
  `wd:Q28920044` (positive integers, i.e. natural numbers excluding 0)

  `wd:Q28920052` (non-negative integers, i.e. natural numbers including 0)

  "Let n be a natural number."
- Will need human aligners
- $\rightsquigarrow$ Idea: Tool supported workflow for efficient alignment

  Inspired by ideas from Snify (Wednesday's talk)

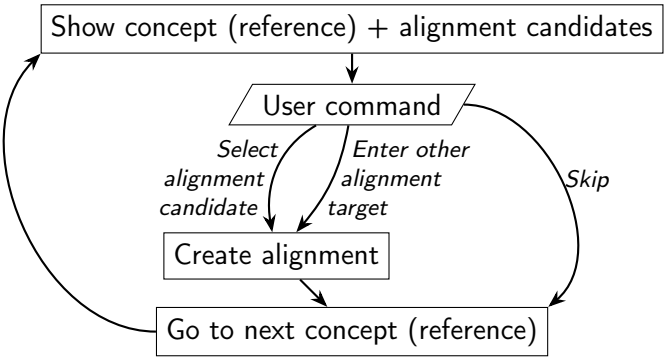
Proposed Workflow



`omp2:E2119`

1. `wd:Q21199`
2. `wd:Q28920044`
3. `wd:Q28920052`

Proposed Workflow



`omp2:E2119`

1. `wd:Q21199`
2. `wd:Q28920044`
3. `wd:Q28920052`

- More information needs to be displayed!
- How do select candidates?

⇝ **Solution: Concept Glossary**

Concept Glossary

Have for each potential alignment target:

- an identifier — *for referencing/making the alignment*
- a (human-readable) description — *so the user knows what the concept is*
- a set of verbalizations — *for linking*

Concept Glossary

Have for each potential alignment target:

- an identifier — *for referencing/making the alignment*
- a (human-readable) description — *so the user knows what the concept is*
- a set of verbalizations — *for linking*

Suggest

`wd:Q28920044`

"*positive integer* (integer greater than zero; natural number explicitly excluding zero)"

"positive integer" (en), "integer greater than zero" (en), "natural number" (en), ...

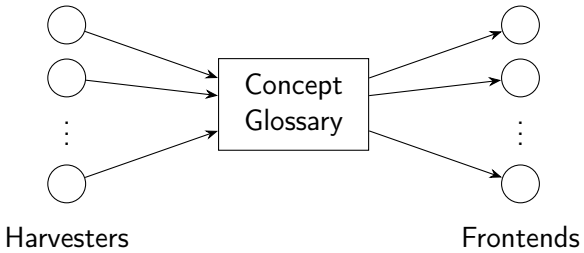
When aligning

`omp2:E2119`: "Natural number" (en), "Натуральное число" (ru)

Find candidates via

- exact verbalization matches
- verbalization matches modulo capitalization/stemming
- something else — *e.g. embeddings*

Harvesters and Frontends



- Different frontends depending on what we align
- Glossary as interface to harvesters
- Harvesters and frontends can be implemented independently and combined freely

kind of

Prototype: Building on Snify

- Snify: tool for efficient sTeX annotation *Presented on Wednesday*
- Today: building prototypes on Snify for WikiData alignment

A WikiData Harvester

Simple SPARQL query to get math concepts *18600 concepts*

```
SELECT DISTINCT ?item WHERE {
    { ?item wdt:P6104 wd:Q8487137 .
        FILTER NOT EXISTS {?item wdt:P31 wd:Q28920044. } .
        FILTER NOT EXISTS {?item wdt:P31 wd:Q5 }.
        FILTER NOT EXISTS {?item wdt:P31 wd:Q10376408 }.
    } UNION {VALUES ?item { wd:Q204 wd:Q199 wd:Q200 }}.
  }
```

Examples:

- "Nonlinear autoregressive exogenous model"
- "line-cylinder intersection"
- "spectral norm"
- "Shannon's expansion"
- "Napoleon-Feuerbach cubic"

Demo 1

Aligning Formulae

WikiData has some notations → add them to glossary

- Identifier: `wd:Q167`
- Description: "constant ratio of the circumference of a circle to its diameter"
- Verbalizations: `\pi` (LaTeX), π (Unicode), "pi" (en), "Archimedes' constant" (en), ...

Aligning Formulae

WikiData has some notations → add them to glossary

- Identifier: `wd:Q167`
- Description: "constant ratio of the circumference of a circle to its diameter"
- Verbalizations: `\pi` (LaTeX), π (Unicode), "pi" (en), "Archimedes' constant" (en), ...

- Can annotate LaTeX formulae
- Can annotate HTML documents (text/MathML formulae) *need HTML interface*
- Limitation: Few notations in WikiData
- Limitation: Can only annotate (some) operators/constants, but not their arguments

Demo 2

Quick Prototype: Aligning OntoMathPro

```
                                    Concept 60/3765
http://ontomathpro.org/omp2#E100

Envelope , Огибающая
http://en.wikipedia.org/wiki/Envelope_%28mathematics%29

Commands:
  [h]elp
  [0] envelope (Q290667): pattern describing a sound or note's changing amplitude over
     time
  [1] envelope (Q1060372): family of curves in geometry
  [s]kip once
>>> 
```

Alignments Extend Glossary

- Text/formula alignment yields new verbalization
- Transport verbalizations/notations across alignments

Example:

1. Align occurrence of `\mathbb{N}` in a document with `wd:Q21199`
2. Align `wd:Q21199` with `omp2:E2119`
3. $\rightsquigarrow$ `\mathbb{N}` is also a notation for `omp2:E2119`

Conclusion

- A simple workflow for efficient alignment
- Separation of harvester and frontend; glossary as interface
- Prototype implementation building on Snify:
  - Can align LaTeX/HTML, both text and formulae, with WikiData
  - Can align OntoMathProV2 with WikiData
  - Only a proof-of-concept, still needs some work
- Where should the alignments be stored?
- Maybe there are other simple ways to help humans align efficiently?

https://github.com/slatex/stextools, `snify2` branch