What Is Needed for Intra-document Disambiguation of Math Identifiers?

Takuto Asakura and Yusuke Miyao, University of Tokyo @ LREC-COLING2024, Turin, Italy

Long-term Goal: P2C Conversion

STEM Documents (Natural Language + Formulae)	
Paper, Textbook, etc.	Calculating the mean of n^2 for $n = 1,2,\ldots,10$

⬇ Conversion

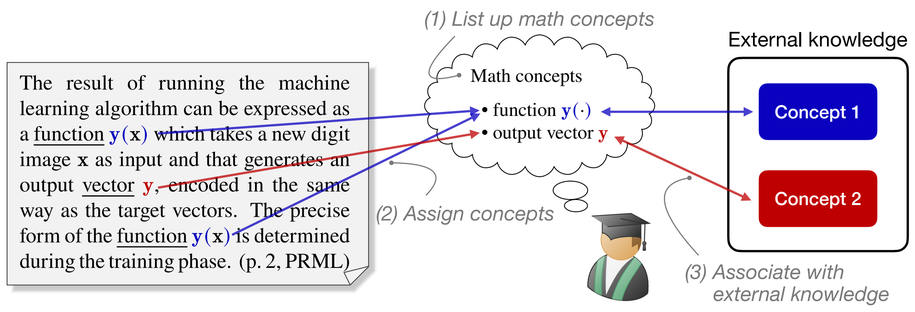
Computational Form (Formal Language)	
Executable code,	`mean([n^2 for n`
First-order logic, etc.	`in range(1, 11)])`

Technologies for the conversion

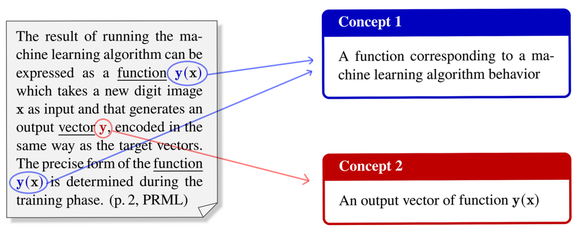
- NL: POS tagging, semantic parsing, text classification, etc.
- Formulae: Token-level analysis, parsing, etc.
- **Integration** of NL texts and formula analyses

Grounding of Formulae

1. List up math concepts used in a document cf. Definition extraction
2. **Assign a math concept** to each math token occurrence
3. Associate math concepts with external knowledge cf. MathIR



Math Concept Assignment



Our Scope

- Problem to solve: **Intra-document ambulation**
- Target: **Math identifiers** (most frequent token type)

Research Questions

- What is the *important feature* for the disambiguation?
- Are those features *depends on domain* of the papers?

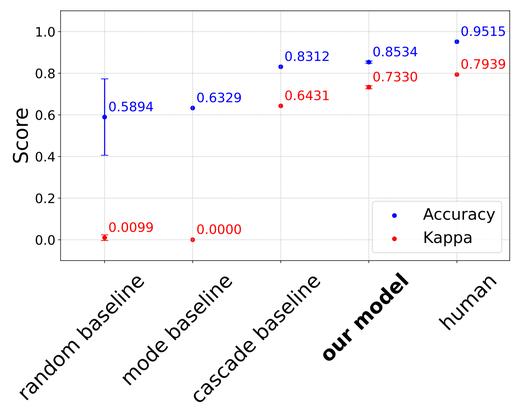
Task Overview

Input
- Structured document representation (XHTML)
- The *initial occurrence location* associated with each math concept for identifier = about 10% of the labels

Output *Math concepts* assigned to every occurrence identifiers = remaining 90% of the labels

Task Difficulty

- Cascade baseline
  = assuming no scope switches occurs except initial pos
  → Kappa **0.6431**
- Human annotators
  → Kappa **0.7939**



Use of Dataset

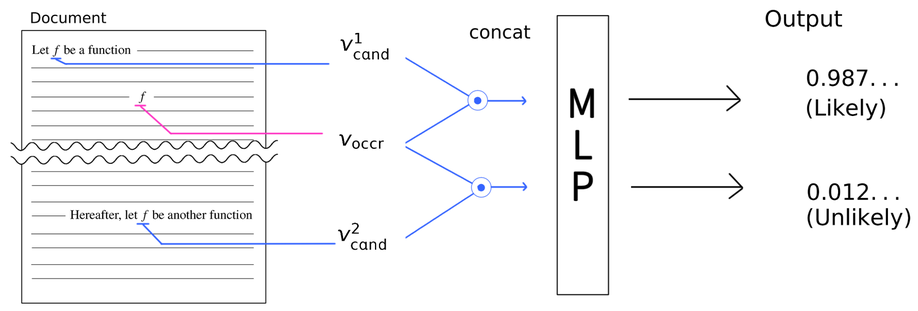
- Split the dataset according to the field of the paper
  - NLP subset: used for both **development and evaluation**
  - Others subset: used only for **evaluation**

Subsets of the Dataset

Subset	#papers	#words	#idf_types	#occrs	#concepts
NLP	20	97,045	789	**9,278**	1,518
Others	20	140,017	953	18,377	2,085
Total	40	237,062	1,742	27,655	3,603

https://sigmathling.kwarc.info/resources/grounding-dataset/

Usage of Multi-Layer Perceptron (MLP)



- The label set (candidate concepts) vary from paper to paper
  → It is not a simple multiclass classification
- The use of MLP is somewhat unique
  1. Make the pairs of (occurrence, concept) for each occurrence
  2. Train MLP to predict the *likelihood of the correct pair*

  → **This method can be used for unknown label sets**

Feature Engineering

c: Context Embeddings

- Natural language text surrounding the target occurrence
  E.g. `feature vector $v'_{x}$ extracted from`
- Vector embeddings with *Sentence Transformer* [Reimers+, 2019]
  - Used MiniLM as a pretrained model (because it performed the best)
  - Impacts of window size and formula representation are little

a: Affix Types

- Local formula structure E.g. Use of sub-/super-script
- We built a rule-based detector → Accuracy 90.56%

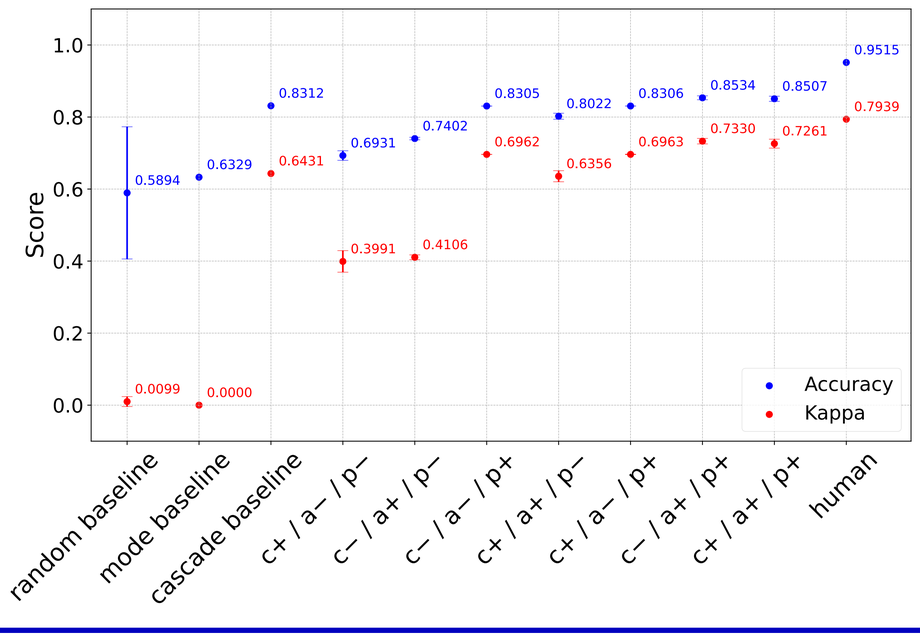
p: Position Data

- Cascade effect scope and distance from the initial position
  → **effective** even if it soley used (identical to cascade baseline)

Model Comparison

We trained our model with various combinations of the features

- Model variations: $2^3 - 1 = 7$ models
- Used features are represented with a letter
  E.g. c+ / a+ / p−: the model using context and affix types



Cross-domain Comparison

We trained our models with NLP subset, and evaluated with others