

A T_EX-oriented Research Topic: Synthetic Analysis on Mathematical Expressions and Natural Language

Takuto ASAKURA

National Institute of Informatics
(Supervisors: Prof. Yusuke Miyao & Prof. Akiko Aizawa)

2019-08-10

A T_EX-driven Life

- ▶ I met T_EX when I was a **high school** student
→ at that time, I'm deeply interested in biology
- ▶ Later, I majored **bioinformatics**—combination of biology & informatics—for my bachelor degree
- ▶ I learned computer science with T_EX

Implementing bioinformatics algorithms in T_EX

The Gotoh algorithm: DP

Sequence alignment has a slightly more complex scoring scheme.

Example

match = 1, mismatch = -1, $g(l) = -d - (l - 1)e$

The algorithm

Sequence alignment in $O(mn)$ time:

$$M_{i+1,j+1} = \max \{ M_{ij}, I_{xij}, I_{yij} \} + c_{a,b}$$

where

$$I_{x+1,j} = \max \{ M_{ij} - d, I_{xij} - e, I_{yij} - d \},$$

$$I_{y_i,j+1} = \max \{ M_{ij} - d, I_{yij} - e \}.$$

5/11

Implementing bioinformatics algorithms in T_EX

The **Gotoh** package

Usage

- ▶ `\Gotoh{(sequence A)}{(sequence B)}`
 - Executes the algorithm
 - Returns the results to specified CSs
- ▶ `\GotohConfig{(key-value list)}`
 - Setting various parameters
 - e.g. algorithm parameters, CSs to store results

Example

Input:

```
\Gotoh{ATCGGGCGCACGGGGGA}
  {TTCCGCCACA}
\texttt{\GotohResultA} \
\texttt{\GotohResultB}
```

Output:

```
ATCGGGCGCACGGGGGA
TTCCGCCACA . . . . .A
```

8/11

An Idea from T_EX: Toward NLP

Representing meanings with T_EX macros

Instead of directly using primitives or standard commands, we can define our own macros which reflect “meanings”.

Example

To express a vector with a **bold** font:

- ✗ Directly writing “ \mathbf{x} ”
- ✓ Defining “ $\def\vector#1{\mathbf{#1}}$ ” and using the macro as “ \vector{x} ”

But: many authors neglect such representation.

How about automating the process?

Targets: STEM Documents

The targets of our work are Science, Technology, Engineering, and Mathematics (STEM) documents.

Example

- ▶ Papers,
- ▶ Textbooks, and
- ▶ Manuals, etc.

STEM documents are:

- ▶ essence of human knowledge
- ▶ well organized (semi-structured)
- ▶ texts with mathematical expressions



Long-term Goal: Converting STEM Documents to Formal Expressions

STEM Documents (Natural Language + Formulae)

Papers, textbooks, manuals, etc.



Conversion

Computational Form (Formal Language)

Executable code, first-order logic, etc.

The conversion enables us to:

- ▶ construct databases of mathematical knowledge
- ▶ search for formulae

Necessity of Synthetic Analysis

Interaction among texts and formulae

Texts and formulae are complimentary to each other:

[Kohlhase and Iancu, 2015]

- ▶ Texts explains formulae (and vice versa)
- ▶ Texts in formulae E.g. $\{x \in \mathbb{N} \mid x \text{ is prime}\}$
- ▶ Notations and verbalizations
E.g. $1 + 2$ and “one plus two”

Deep synthetic analyses on natural language and mathematical expressions are necessary.

Grounding Elements to Mathematical Objects

- ▶ Elements in formulae and their combination can refer to **mathematical objects**
- ▶ The detection is fundamental for understanding STEM documents

Example

For example, x might describe the outcome of flipping a coin, with $x = 1$ representing 'heads', and $x = 0$ representing 'tails'. We can imagine that this is a damaged coin so that the probability of landing heads is not necessarily the same as that of landing tails. The probability of $x = 1$ will be denoted by the parameter μ . The probability distribution over x can therefore be written in the form

The probability of 'heads' on top, float, $0 \leq \mu \leq 1$

$$\text{Bern}(\overset{\uparrow}{\mu} \mid \underset{\downarrow}{x}) = \mu^x(1 - \mu)^{1-x}$$

The result of coin flipping, int, $x \in \{0, 1\}$

which is known as the *Bernoulli* distribution. (PRML, pp. 86–87)

Difficulty of the Grounding

Factors which make the detection highly challenging:

- ▶ ambiguity of elements (see below)
- ▶ syntactic ambiguity of formulae **E.g.** $f(a + b)$
- ▶ necessity for common sense & domain knowledge
- ▶ severe abbreviation

Usage of character \mathbf{y} in the first chapter of PRML (except exercises)

| Text fragment from PRML Chap. 1 | Meaning of \mathbf{y} |
|---|--|
| ... can be expressed as a function $\mathbf{y}(\mathbf{x})$... | a function which takes an image as input |
| ... an output vector \mathbf{y} , encoded in ... | an output vector of function $\mathbf{y}(\mathbf{x})$ |
| ... two vectors of random variables \mathbf{x} and \mathbf{y} ... | a vector of random variables |
| Suppose we have a joint distribution $p(\mathbf{x}, \mathbf{y})$... | a part of pairs of values, corresponding to \mathbf{x} |

Semantics Over Natural Language and Mathematical Expressions

There are ambiguity arise only when **context** exists. For instance, “equals signs” (=) in formulae have at least three meanings: **definition**, **identity**, and **equation**.

Example

Let $a \stackrel{\text{def}}{=} 4$, $b \stackrel{\text{def}}{=} 3$. Suppose we have to solve

$$ax^4 + bx^2 + 1 \stackrel{\text{eq}}{=} 0.$$

To reach the answer, “difference of two” is helpful:

$$p^2 - q^2 \stackrel{\text{id}}{=} (p + q)(p - q).$$

Dataset arXMLiv

- ▶ papers from arXiv in XML format [Ginev+, 2009]
 - ▶ converted from \LaTeX via \LaTeXML
 - ▶ formulae are in MathML markups

XHTML/XML

III-B Defining Supervised Learning

Having introduced the goal of supervised learning, we give a formal definition of the problem. Throughout this section, x and t are random variables and the corresponding letter

As a starting point, we assume that the training set

$$(x_n, t_n)_{i.i.d.} \sim p(x, t), \quad n = 1, \dots, N, \quad (1)$$

that is, each training sample pair (x_n, t_n) is generated independently (i.i.d.) from a joint distribution $p(x, t)$ and the sample pairs are independent (i.i.d.). As discussed, based on the training set \hat{S} , we seek a predictor $\hat{f}(x)$ that performs well on any possible relevant task formalized by imposing that the predictor is $\hat{f}(x, t) \sim p(x, t)$, which is generated independently \mathcal{D} .

The quality of the prediction $\hat{f}(x)$ for a test pair (x, t) is measured by a given loss function $\ell(t, \hat{f})$ as $\ell(t, \hat{f}(x))$. Typical examples of loss functions include the quadratic loss $\ell(t, \hat{f}) = (t - \hat{f})^2$ for regression problems; and the error rate $\ell(t, \hat{f}) = 1(t \neq \hat{f})$, which equals 1 when the prediction is incorrect, i.e., $t \neq \hat{f}$, and 0 otherwise, for classification problems.

The screenshot shows the arXiv.org interface for a paper in the Computer Science > Information Theory category. The paper title is "A Very Brief Introduction to Machine Learning With Applications to Communication Systems" by Ovidio Simeone. It was submitted on 7 Aug 2018 and revised on 5 Nov 2018. The abstract discusses the challenges of applying data-driven machine learning methods to communication systems. The page includes a download section with PDF and other formats, a current browse context, and a list of references and citations.



A Little Note for MathML

- ▶ a W3C Recommendation [Ausbrooks+, 2014]
- ▶ includes two markups: presentation and content

Presentation Markup

This shows syntax:

```
<msup>
  <mfenced>
    <mi>a</mi>
    <mo>+</mo>
    <mi>b</mi>
  </mfenced>
  <mm>2</mm>
</msup>
```

Content Markup

This shows semantics:

```
<apply>
  <power>
    <apply>
      <plus/>
      <ci>a</ci>
      <ci>b</ci>
    </apply>
    <cn>2</cn>
  </apply>
```

$$(a + b)^2$$

The Research Plan

Creating a dataset (pilot annotation)

- ▶ do the grounding **by hand** for some papers in arXiv
→ Let me show you a **demonstration**
- ▶ I would also like to do it for some textbooks

Automating the detection

Combination of rule-based and machine learning with features such as:

- ▶ **apposition nouns** E.g. “a **function** *f*”
- ▶ **syntactic information** in formulae
E.g. does it appear inside an argument or not?
- ▶ **distance** from the former appearance

Possible Applications

- ▶ Mathematical Information Retrieval (MIR)
→ enables us to create scientific knowledge bases
- ▶ Automatic code generation E.g. Python, Coq, etc.
- ▶ Searching for mathematical expressions

Example

Let us think about searching for:

$$x^n + y^n = z^n \quad (n \geq 3).$$

It is easy to search if you know a keyword *Fermat's Last Theorem*, but otherwise. . .

Conclusions

- ▶ converting STEM documents to computational form is beneficial and challenging
- ▶ for the conversion, **synthetic analysis** on natural language and mathematical expressions is required
- ▶ Currently, we are working on creating a dataset
- ▶ Possible applications: MIR, code generation, searching for formulae

T_EX has a power to change one's life!