A \TeX-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 \TeX-driven Life

▶ I met \TeX 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 \TeX

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:

\[
M_{i+1,j+1} = \max\{M_{ij}, I_{xij}, I_{yij}\} + c_{a_ib_j}
\]

where

\[
I_{x_{i+1,j}} = \max\{M_{ij} − d, I_{xij} − e, I_{yij} − d\},
I_{y_{i,j+1}} = \max\{M_{ij} − d, I_{yij} − e\}.
\]

The \texttt{Gotoh} package

**Usage**

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

**Example**

Input:

\[
\texttt{\Gotoh\{ATCGGGCGACGCGGGA\}\TTCCGCACACA}
\]

Output:

\[
\texttt{ATCGGGCGACGCGGGA}
\texttt{TTCCGCACACA . . . A}
\]
An Idea from \TeX: Toward NLP

Representing meanings with \TeX 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 \textbf{bold} font:
- \times Directly writing “$\mathbf{x}$”
- \checkmark Defining “\texttt{\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 $\mu$. The probability distribution over $x$ can therefore be written in the form

$$\text{Bern}(x | \mu) = \mu^x (1 - \mu)^{1-x}$$

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

<table>
<thead>
<tr>
<th>Usage of character ( y ) in the first chapter of PRML (except exercises)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Text fragment from PRML Chap. 1</strong></td>
</tr>
<tr>
<td>… can be expressed as a function ( y(x) ) …</td>
</tr>
<tr>
<td>… an output vector ( y ), encoded in …</td>
</tr>
<tr>
<td>… two vectors of random variables ( x ) and ( y ) …</td>
</tr>
<tr>
<td>Suppose we have a joint distribution ( p(x, y) ) …</td>
</tr>
</tbody>
</table>
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 \( a = 4 \), \( b = 3 \). Suppose we have to solve

\[
ax^4 + bx^2 + 1 = 0.
\]

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

\[
p^2 - q^2 = (p + q)(p - q).
\]
Dataset arXMLiv

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

XHTML/XML

III-B Defining Supervised Learning

Having introduced the goal of supervised learning, we now proceed to a formal definition of the problem. Throughout, we will use random variables and the corresponding lettering.

As a starting point, we assume that the training data is a sample from a distribution \(p(x, t)\), where \(x\) is the input and \(t\) is the target value.

\[
(x_n, t_n) \sim p(x, t), \quad n = 1, \ldots, N,
\]

that is, each training sample \((x_n, t_n)\) is drawn from the distribution \(p(x, t)\) and the sample pairs are i.i.d. As discussed, based on the training set, we are interested in finding a classifier \(\hat{t}(x)\) that performs well on any possible relevant test set \(\mathcal{D}\).

The quality of the prediction \(\hat{t}(x)\) for a test pair \((x, t)\) is measured by a given loss function \(\ell'(t, \hat{t}(x))\). Typical examples of loss functions include the quadratic loss \(\ell'(t, \hat{t}) = (t - \hat{t})^2\) for regression problems; and the error rate \(\ell'(t, \hat{t}) = 1(\hat{t} \neq t)\), which equals 1 when the prediction is incorrect, i.e., \(t \neq \hat{t}\), and 0 otherwise, for classification problems.
A Little Note for MathML

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

**Presentation Markup**
This shows syntax:

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

**Content Markup**
This shows semantics:

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

**\( \text{T\řX has a power to change one’s life!} \)**