Machine translation of mathematical text (using LaTeX)

Aditya Ohri [1] Tanya Schmah [2]

[1] EECS, University of California, Berkeley, aditya.ohri@berkeley.edu

[2] Math & Stats, University of Ottawa, tschmah@uottawa.ca

TeX Users Group meeting, July 24th 2022

Outline

- The Problem
- The Workarounds
- A Solution: the PolyMath Translator [presented by Aditya]
- Recent developments [presented by Tanya]
- The Future

The Problem

Translate a mathematical document (in LaTeX, naturally)
into a different natural language, e.g. from English to French.

written $J^k_x(M,N)$. The set $J^k(M,N)$ is the union of these sets, for all x. It is a smooth vector bundle over $M \times N$, called the *k-jet bundle*. The k-jet of f with source x is written $j^k f(x)$. In local coordinates, $j^k f(x)$ "is" the k^{th} order Taylor expansion of f at z. The *k-jet extension* of $f : M \to N$ is the map

$$j^k f : M \longrightarrow J^k(M,N)\,; \qquad x \longmapsto j^k f(x)\,.$$

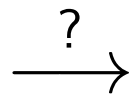
If f is smooth, then $j^k f$ is as well, so there is a map

$$j^k : \mathcal{C}^\infty(M,N) \longrightarrow \mathcal{C}^\infty\left(M, J^k(M,N)\right), \tag{1}$$

taking f to $j^k f$. This map is continuous, with respect to the strong topologies on domain and codomain [GG73].

Theorem 2.5 (Jet transversality). *Let M and N be manifolds and let S be a submanifold of $J^k(M,N)$. Then the set of functions $f : M \to N$ such that $j^k f$ is transverse to S is residual in $C^\infty_s(M,N)$, and open dense if S is closed.*

To apply jet transversality to vector fields, we need the modified version in Theorem 2.6. This result is known, but we are unaware of a proof in the literature. We will prove it from Theorem 2.5, using the globalisation technique in Lemma 2.8. We will re-use the same globalisation lemma in the proof of the new result in Theorem 2.9.



The Workarounds

Translate a mathematical document.

- LaTeX $\xrightarrow[\textit{DeepL?}]{\textit{GoogleTranslate?}}$ (not LaTeX)

Original English LaTeX (compiled)	**Definition 2.1.** Let $x \in \mathbb{F}^n$. The *closed ball of radius* r *centered at* x is $S_r(x) = \{y \in \mathbb{F}^n \mid d(x,y) \leq r\}$.

Google Translate (results uncompilable)	`\ begin {defn}`
	`Soit $ x \ dans \ F ^ n$.}`
	`La \ define {boule ferme de rayon $ r $`
	`centre sur $ x $} est`
	`$$`
	`S_r (x) = \ {y \ in \ F ^ n \ mid d (x, y) \ leq r \}.`
	`$$`
	`\ end {defn}`

The Workarounds

Translate a mathematical document.

- LaTeX $\xrightarrow[\text{DeepL ?}]{\text{Google Translate ?}}$
- PDF $\xrightarrow{\text{some PDF reader}}$ plain text $\xrightarrow{\text{Google Translate}}$ plain text
- LaTeX $\xrightarrow[\text{DeepL}]{\text{Google Translate API}}$
  using a customized glossary to avoid translation of reserved words such as `\in`
  This last solution sort of works, but takes time and expertise; and it can't take advantage of any LaTeX semantics.

A Solution: the PolyMath Translator

A machine translation system (currently English-French) for LaTeX documents containing mathematical text.

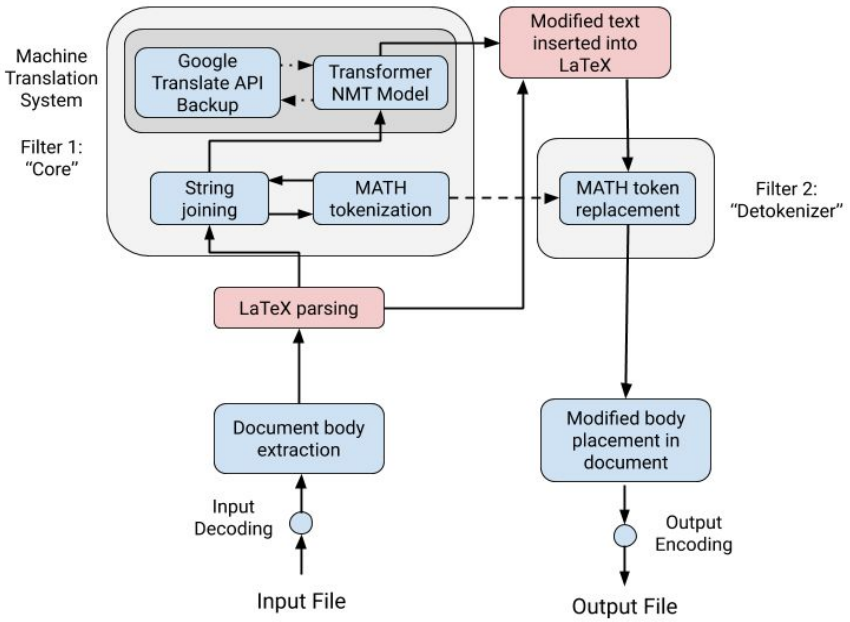


FIGURE 2. Overview of PolyMath Translator. The pink-coloured modules are implemented using the pandoc universal document converter.

LaTeX Parsing

We use the Pandoc Universal Document Converter (Python wrapper pypandoc) to convert a document from LaTeX to LaTeX, leveraging intermediate internal document representation: JSON-formatted *abstract syntax tree* (AST).

English LaTeX: `Let $Y$ ...`

pandoc ↓

Abstract Syntax Tree:

```
[{''t'':''Str'',''c'':''Let''\},
{''t'':''Space''},{''t'':''Math'',''c'':
:[{''t'':''InlineMath''},''Y'']},...
```

filters ↓

Abstract Syntax Tree:

```
[{''t'':''Str'',''c'':''Soit''},
{''t'':''Space''},{''t'':''Math'', ''c'':
[{''t'':''InlineMath''},''Y'']},...
```

pandoc ↓

French LaTeX: `Soit $Y$ ...`

Filter 1: “Core”

- Limitation of AST: individual inline tokens separated → poor translation quality
- Solution: two-layered filter
  - Manipulating individual block elements through Pandoc interface
  - “String-joining” function to manipulate inline elements directly (Python)
    - Combines inline tokens into complete sentences (represented as a string element) including math formulas tokenized as “MATHnX”
    - Full sentences then translated (more in next slides) and placed back into AST by Pandoc filter

“Let Y have mean μ and variance σ^2, and an unknown p.d.f. p_Y that is everywhere nonzero.”

[“t”:“Str”,“c”:“Let MATH1X have mean MATH2X and variance MATH3X, and an unknown p.d.f. MATH4X that is everywhere nonzero.”]

Filter 2: “Detokenizer”

- Now, we have a modified LaTeX document with all natural language in French and “MATH” tokens
- Replaces “MATHnX” tokens with corresponding original mathematical formulas
  - Uses n-index to retrieve corresponding formula from JSON object created by “Core” filter containing saved math formulas
- End result: document AST contains French text and original math expressions; Pandoc then creates new LaTeX file based on this internal representation

Machine Translation System: Transformer

- Trained custom neural machine translation model (NMT) using Transformer architecture
  - "Sequence-to-sequence" translator implemented using a neural network
  - For any position t in the output sequence y, model outputs a conditional distribution $p(y_t \mid y_{<t}, x)$ based on the entire input sequence x and the pre- ceding outputs $y_{<t}$
  - Key distinguishing feature: Multi-Head Self-Attention
    - Attention function applies weights to elements of input sequence depending on position t in output sequence
    - Multi-head attention applies several parallel attention functions, seeing input from different "points of view"
      - Ex: I kicked the ball. *verb conjugation; verb root*
    - Especially useful in mathematical domain (grammatical structures)

Machine Translation System: Training

- No published corpora for evaluating mathematical translation; created own corpus of three components, all containing parallel English-French sentences:
  - Subset of OPUS “Wikipedia” corpus, applying naive subject matter filter of at least 2 terms from custom math glossary — 16,767 sentence pairs
  - Subset of “Aligned Hansards of the 36th Parliament of Canada” — 250,000 sentence pairs
    - Debates from House of Commons/Senate; formal expository style of language with similar structure to mathematical text
    - Greater breadth of vocabulary, grammar, style
  - “Custom math” corpus: Schmah’s math research papers, manually translated — 1,075 sentence pairs

Machine Translation System: Backup

- Google Translate with custom math glossary if Transformer produced output sentence with confidence lower than threshold
  - Threshold tuned to maximize BLEU on validation set
  - Useful for general English text; model may not perform as well as commercial translation services (limited training set)
- Reference point: NMT model used 71%; GT used 29% during validation

Results

	main corpus (multi-domain)	linear code corpus (mathematical)
Full PM	**32.5**	**53.6**
PM-Transformer	29.0 *(-3.5)*	50.4 *(-3.2)*
PM-Google	27.7 *(-4.8)*	46.5 *(-7.1)*
PM-Piece	—	38.0 *(-15.6)*
Google Raw	—	31.6 *(-22.0)*

Summary of PolyMath Translator Results from 2021

Successful initial implementation using `pandoc`:

- Excellent translation quality (BLEU 53.5 on small test corpus).
- Output is LaTeX that *usually* compiles without hand-correction.
- Moderate ease-of-use.

A Ohri, T Schmah (2021) Machine translation of mathematical text.
IEEE Access 9, 38078-38086

Some limitations of original PolyMath Translator:

- Only supports English to French.
- Only translates single files.
- No user configuration or editable glossaries
- Doesn't use LaTeX semantics to e.g. *not* translate verbatim environments, or comments, or certain arguments.

A Ohri, T Schmah (2021) Machine translation of mathematical text. IEEE Access 9, 38078-38086

Some limitations of original PolyMath Translator:

- Only supports English to French.
- Only translates single files.
- No user configuration or editable glossaries
- Doesn't use LaTeX semantics to e.g. *not* translate verbatim environments, or comments, or certain arguments.

- changes the LaTeX commands

A Ohri, T Schmah (2021) Machine translation of mathematical text. IEEE Access 9, 38078-38086

A limitation of original PolyMath Translator :

It changes the LaTeX commands.

For example, `\textit{hello}`
is translated to `\emph{bonjour}`
instead of `\textit{bonjour}`.

This is an inevitable consequence of using `pandoc` to translate LaTeX into an abstract internal representation and then back into LaTeX.

This may be acceptable for some applications, e.g. browsing articles, but not for e.g. book authors.

A limitation of original PolyMath Translator :

It introduces errors into some LaTeX commands.

For example, `\includegraphics[scale=0.2]{file.png}`
is translated to `\includegraphics{file.png}`

This is again a consequence of using `pandoc` to "translate" from LaTeX to LaTeX:
Since `pandoc` is trying to *interpret* and *translate* every LaTeX command, it will always fail on commands it doesn't know.

From the point of view of PolyMath, `pandoc` is trying to do too much: it's trying to understand the semantics of LaTeX when we mainly just need the syntax.

What a translator needs to understand about LaTeX

Mainly the syntax.

Plus, enough semantics to . . .

- identify which arguments should be translated, e.g.:
  - Don't translate: math; label names;
  - Do translate: title, section names, text mode strings inside math environments;
- tokenize math expressions (inline and displayed environments);
- tokenize label references;
- translate other files referred to in `\input` and `\include` commands.

PolyMath Translator v0.2-dev, using TexSoup parser

TexSoup

downloads 9.1k/month | build passing | coverage 100%

TexSoup is a fault-tolerant, Python3 package for searching, navigating, and modifying LaTeX documents.

Created by Alvin Wan + contributors.

Inspired by `Beautiful Soup`, a Python package for parsing HTML and XML documents.

https://github.com/alvinwan/TexSoup
https://texsoup.alvinwan.com

PolyMath Translator v0.2-dev, using TexSoup parser

This version of PolyMath leaves TEX commands unchanged, and understands *just enough* semantics. It includes:

- Editable lists of which command and environment arguments to translate
- Tokenization of math expressions
- Tokenization of label references
- Translation of entire file trees using `\input` and `\include`.

Experience with the PolyMath Translator at uOttawa

- Automatic translation
  - A textbook for Intro to Math Models, which was automatically translated for a francophone student in an English-language class.
- Semi-automated translation (automatic + post-editing)
  - Course notes for Intro to ODEs and Multivariable Calculus

Example 2.1 Résoudre le problème de la problème à valeur initiale

$$y'(t) + 2y(t) = 10, \qquad y(0) = 1.$$

Cette équation se présente sous la forme (2.1) avec $p(t) = 2$ et $g(t) = 10$. On multiplie par une fonction $\mu(t)$ et on obtient

$$\mu(t)y'(t) + \underbrace{2\mu(t)}_{\mu'} y(t) = 10\mu(t).$$

Ainsi, l'équation définissant le facteur intégrant est $\mu'(t) = 2\mu(t)$, qui a une solution $\mu(t) = \exp(2t)$. En multipliant l'équation par le facteur intégrant,

Experience with the PolyMath Translator at uOttawa

Findings:

- PolyMath produces excellent translations that nonetheless need correction and polishing.
- The uncorrected automatic translation is already useful.
- Overall, a semi-automated professional translation process is about twice as fast as manual translation.
- PolyMath, and the TexSoup parser, still have issues and require an expert user.

In progress: the Ottawa Mathematical Term Bank

Machine translation makes heavy use of subject-specific *glossaries*, i.e. *dictionaries*.

A term bank is a like a glossary but more specific, containing information to disambiguate homonyms:

field	algebra	corps	Körper
field	database	champ	Datenfeld
finitely generated group		groupe de type fini	endlich erzeugte Gruppe

From a term bank, a glossary can be extracted and customised for each project.

“Translating mathematical text” vs. “Translating LaTeX documents”

Most of the work in this project applies to any LaTeX document.

Some of it is math-specific:

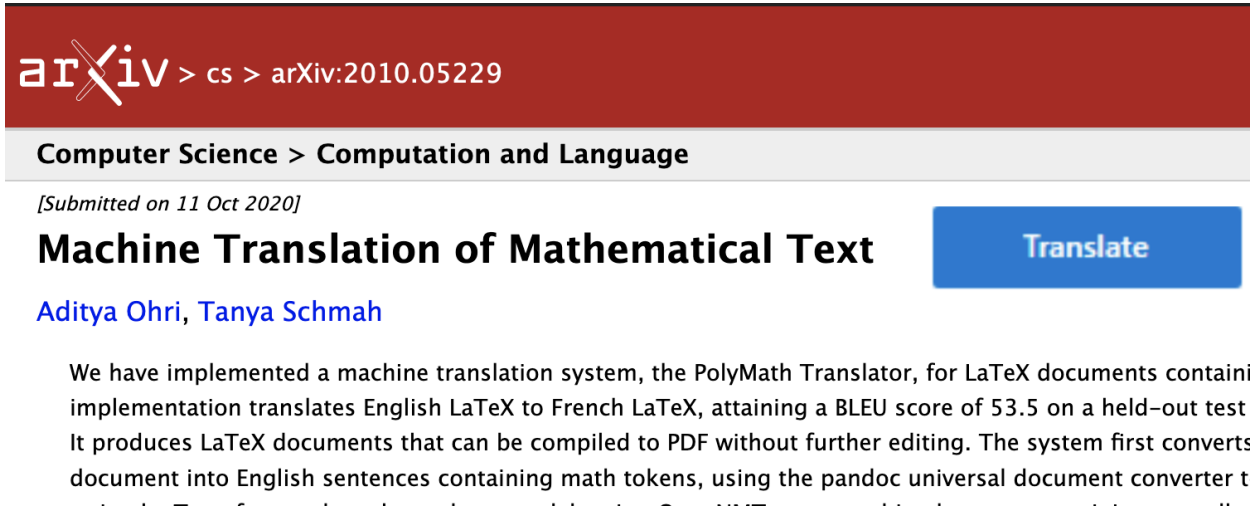
- Tokenization of math expressions (inline and displayed).
- Development of training corpora of mathematical sentence pairs (in English and French).
- Training of neural networks on math-heavy corpora.
- The Ottawa Mathematical Term Bank.

In general, domain-specific translators (e.g. in law and medicine) have given better results than general-purpose ones. Our work suggests that this will be true for mathematical text as well.

This is a big opportunity for the math community to improve communication, accessibility and inclusion.

A vision of the future

- An open source PolyMath Translator project, supported by:
- Open data: term banks, training corpora, and pre-trained deep learning models for many language pairs
- Improved math-specific translation, e.g. using content of math expressions, or using topic models to select appropriate translations for ambiguous terms.
- A web service (like "Google Translate for LaTeX")
- A “translate” button in preprint servers.



Contributions welcome!

Especially:

- Glossaries (or term banks) for many language pairs and many specialized subjects.
- Training data: pairs/sets of corresponding sentences in different languages.
- Technical advice:
  - choice of LaTeX parser; if TexSoup, further contributions to that project;
  - setting up an open source project;
  - Hosting, naming, structuring term banks and training corpora.

Thank you!