Understanding scientific documents with synthetic analysis on mathematical expressions and natural language*

Takuto Asakura

1 Introduction

Converting Science, Technology, Engineering, and Mathematics (STEM) documents to formal expressions has a large impact on academic and industrial society. It enables us to construct databases of mathematical knowledge, search for formulae, and develop a system that generates executable code.

However, such conversion is an exceedingly ambitious goal. Mathematical expressions are commonly used in scientific communication in numerous fields such as mathematics and physics, and in many cases, they express key ideas in STEM documents. Despite the importance of mathematical expressions, formulae and texts are complementary to each other, and neither can be understood independently. Thus, deep synthetic analyses on natural language and mathematical expressions are necessary.

To date, much effort has been made for developing Natural Language Processing (NLP) techniques, including semantic parsing (SP) [4], but their targets are mostly 'general' texts. Naturally, conventional NLP techniques include only limited features to treat formulae and numerous linguistic phenomena specific to STEM documents [3].

Meanwhile, semantics on mathematical expressions also has been deeply investigated. Such results can be seen in logic theories, the MathML specification [1], etc. However, there is a large distance between formal expressions such as first-order logic and actual formulae in natural language texts.

2 Research goals

There is substantial work remaining to achieve conversion from STEM documents to a computational form (Figure 1). At first, we are going to focus on the two foundational parts for the synthetic analyses. The first is token-level analyses on formulae. The main part of the analyses is associating formulae tokens to mathematical objects and text fragments (Section 2.1). This is a fundamental step for the conversion, but it is still almost untouched. The second step is the morphology of mathematical expression and semantics covering both formulae and texts (Section 2.2). Studying underlying theories is essential to deeply understand the structure of STEM documents. We aim for practical applications via a bottom-up approach.

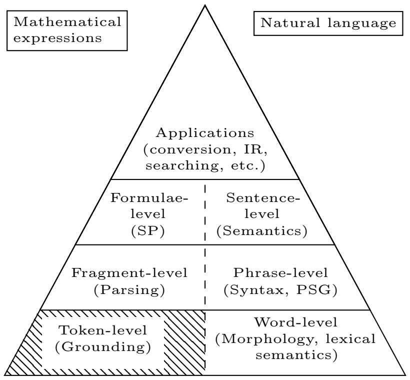
Mathematical expressions
Natural language
Applications (conversion, IR, searching, etc.)
Formulae-level (SP)
Sentence-level (Semantics)
Fragment-level (Parsing)
Phrase-level (Syntax, PSG)
Token-level (Grounding)
Word-level (Morphology, lexical semantics)

Figure 1: Overview of our task definitions. At first, we are tackling the token-level analyses on mathematical expressions (Section 2.1) and theories covering both formulae and texts (Section 2.2).

2.1 Associating tokens in formulae with mathematical objects and their descriptions in texts

Tokens in formulae (e.g., x, ε, $\times$, $\log$) and their combinations can refer to *mathematical objects*. We human beings are able to detect what each token or combination pointing to, by using common sense, domain knowledge, and referencing descriptions in the document or in the others. This detection is fundamental and should be one of the initial steps for understanding STEM documents, but unfortunately, it cannot be easily done by a machine. There are at least four factors which make the detection highly challenging: (1) ambiguity of tokens, (2) syntactic ambiguity of formulae, (3) need for "common" sense and domain knowledge, and (4) severe abbreviation. These difficulties often appear in formulae; giving an example for (1) as a representative, already in the first chapter of a book *Pattern Recognition and Machine Learning* (PRML) [2], a character $\mathbf{y}$ (letter 'y' in bold roman) is used with several meanings including a function, vectors, and a value (Table 1).

The other part of the initial steps of understanding STEM documents is connecting text fragments to the subjective mathematical objects. Our hypothesis is that for this step, general NLP approaches such as dependency parsing are more or less applicable. Of course, some tuning for STEM documents will be required. Also, this process might need to be done by considering the result of mathematical object detection for formulae.

2.2 Semantics and morphology

Semantics on natural language and mathematical expressions have been studied separately. However, to understand STEM documents, it is important to

[*] A version of this extended abstract was published at the Doctoral Programme of CICM 2019.

Table 1: Usage of character $\mathbf{y}$ in the first chapter of PRML (except exercises). Underlines by the author.

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

investigate a synthetic semantics covering both of texts and formulae.

Though morphology has been studied for natural languages, this is not so much the case for formulae. As a matter of fact, in terms of morphology, *words* also exist in formulae. For instance, a token M is a word in "Matrix M", but M is not a word in "An entry $M_{i,j}$" ($M_{i,j}$ is a word). Unlike morphemes in natural language, tokens in formulae do not have lexical categories, but some symbols (e.g., parentheses and equal sign) and positional information (e.g., super/subscript) have typical usages.

3 Completed and remaining research

For the beginning of our research, we simplified the detection task which we described in Section 2.1. Specifically, we are making annotations on some research papers in the following manner:

1. Detecting minimal groups of tokens (we call them *chunks*), each referring to a mathematical object (chunking).
2. Categorizing chunks by the mathematical object they referring to.

This annotation (*pilot annotation*) is the fundamental process for creating the first gold dataset for associating tokens and mathematical objects. The annotated data will also be helpful for investigating the morphology on mathematical expressions.

In other words, we defined a classification task before annotating descriptions for formulae tokens. Since there are many ways to describe a mathematical object, this classification can be done more coherently through the pilot annotation. Moreover, we are expecting that the classification is naturally rather easier to be automated than giving descriptions automatically in the first attempt.

Besides the pilot annotation, all the jobs that have to be done to achieve our goal remain. For the next step, we are planning to automate the annotation process by using features such as apposition nouns and syntactic information in formulae. At the same time, we have to decide the form of mathematical objects. For now, we can say that every mathematical object should have a description and some attributes such as types (e.g., int and float). What attributes are necessary and sufficient is still not clear, and we will find out after trying the annotation for several documents.

4 Publication plans and evaluation plans

Currently, we are creating a new language resource as the pilot annotation, and we are planning to publish it for the community of language resources. For the further future, we will develop automation algorithms for mathematical object detection, which are works suitable for NLP and the digital mathematical library community, including CICM. The analyses on underlying morphology and semantics are more like works in computational linguistics.

For the initial dataset, it is better to ensure agreement among a few experts, if possible. Subsequent progress on developing algorithms and analyses on linguistic phenomena should be evaluated with our handmade gold datasets.

Acknowledgments. I would like to thank my supervisors Prof. Yusuke Miyao (the University of Tokyo) and Prof. Akiko Aizawa (National Institute of Informatics) for their most helpful advice. This work was supported by JST CREST Grant Number JPMJCR1513, Japan.

References

[1] R. Ausbrooks, S. Buswell, et al. *Mathematical Markup Language (MathML) 3.0 Specification.* World Wide Web Consortium (W3C), 2014.

[2] C. M. Bishop. *Pattern Recognition and Machine Learning.* Springer, 2006.

[3] M. Kohlhase and M. Iancu. Co-representing structure and meaning of mathematical documents. *Sprache und Datenverarbeitung, Intl. J. for Language Data Processing* 38(2):49–80, 2014.

[4] S. Reddy, O. Täckström, et al. Transforming dependency structures to logical forms for semantic parsing. *Transactions of the Association for Computational Linguistics* 4:127–140, Dec. 2016.

◇ Takuto Asakura
The Graduate University for
Advanced Studies, SOKENDAI
Department of Informatics
Chiyoda, Tokyo, Japan
tkt.asakura (at) gmail dot com