Applications of Computer Algebra – ACA 2024
June 24-28, 2024
Session on *"Machine Intelligence and Symbolic Computation"*

Formal Representation of Mathematical Texts

Jiang Dongchen[1]***, Hou Yiming***[1] [jiangdongchen@bjfu.edu.cn]

[1] School of Information Science and Technology, Beijing Forestry University, Beijing, China

Existing large language models for mathematics are usually trained by using descriptions of natural languages [1]. However, logical relations between mathematical concepts are not well processed by current models, and it would be proper to train mathematical large language models by some kinds of formal languages.

Currently, the formalization of mathematical text is scarce [2] and typically targets on specific areas. For example, Pan Lu et al. [3] converts the natural language description of geometry into a formal one which contains the basic geometric concepts of point, line and face. Danqing Huang et al. [4] converts applied mathematics textual descriptions into algebraic expressions to obtain algebraic relationships between mathematical concepts. In order to processing more general mathematics descriptions, and to obtain more formal mathematical descriptions for mathematical large language processing, this paper proposes a method that can automatically convert mathematical texts into formal descriptions while keeping the original logical relationships.

Specifically, a predicate-logic based formal language is proposed to character logical relations between mathematical concepts. Based on this language, a standardized syntax tree is used to store and represent relevant entities where all information is obtained by analyzing the parsing trees of the natural language descriptions, and the parsing trees are obtain by using the Stanford CoreNLP tool.

In this paper, the general form of predicate is defined as $P ::= V_{\textbf{node}}(N_{\textbf{node}}|P)^+$, where $V_{\textbf{node}} ::= V < posV > [AV < posAV >]^*[ADV]^*$, $N_{\textbf{node}} ::= N[ADJ|P]^*$. In the predicate, V is the base form of the corresponding verb, AV represents the possible auxiliary verbs, $posX$ denotes the part-of-speech tag of X, ADV represents adverbs, N denotes nouns, and ADJ represents adjectives. If a natural language description is a sentence with adverbial clauses or compound sentences, logical connectors, such as $\neg$,$\wedge$,$\vee$, $\rightarrow$, $\leftrightarrow$, $\circ$, $\square$, are used to link multiple predicates. The logical connectors $\circ$ and $\square$ represent temporal and contrastive relationships. Taks the sentence "If the left and right limits are not equal, then the limit does not exist" as an example, the sentence is converted into the formal form of $\{\neg$be<VBP>(the limits[left $\wedge$ right], equal)$\} \rightarrow \{\neg$exist<VB>[do<VBZ>](the limit)$\}$, and a simplified representation $\{\neg$be(the limits[left $\wedge$ right], equal)$\} \rightarrow \{\neg$exist(the limit)$\}$ can be used for specific processing.

To obtain the form description, the Stanford CoreNLP tool is used to generate the syntactic parse trees of a natural language sentence. The parsing tree is constructed based on Chomsky's generative grammar, which contains all part-of-speech information, and an example is shown in Figure 1. However, the structure and information are not good enough to represent the logic relations between mathematical concepts, and it is still necessary to convert it into a more suitable structure.

In this paper, Hallidays functional grammar is used to construct the structure of a standardized syntax tree, and the final form is also determined by the structure of the formal predicate language. As the example shown in Figure 2, a standardized syntax tree may contain VerbNode, NounNode, CcNode, AdjNode, and AdvNode as its nodes, where VerbNode is used to store the verb-related information and NounNode stores noun-related information.

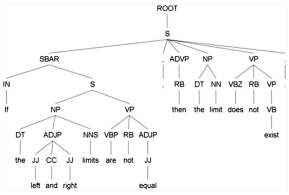

Figure 1: syntactic parse tree

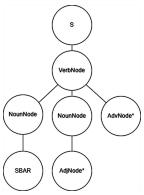

Figure 2: standardized syntactic tree

The conversion from a syntactic parsing tree to the corresponding standardized syntactic tree involves subordinate clause analyzing and sentence component analyzing. After the standardize tree is obtained, the formal predicate description can be printed directly.

To verify correctness of the whole process, 500 sentences from the textbook *CALCULUS* by JAMES STEWART is tested. The correctness of the conversion is evaluated by whether the logical relations and predicate relations are kept. And all results were evaluated by graduate students. Experiment showed that 471 sentences were successfully converted into formal descriptions, which means an accuracy rate of 94.2%.

This automated conversion method can be used to generate a large amount of formal mathematical descriptions from existing mathematical texts. In future work, we would like to use the generated formal descriptions for context inference and mathematical large language model training.

Keywords
Mathematical text, Formal language, Mathematical large language models

References

[1] AHN J; VERMA R; LOU R ET AL, *Large language models for mathematical reasoning: Progresses and challenges*. *arXiv preprint* arXiv:2402.00157 (2024).
[2] WU Y; JIANG A Q; LI W ET AL, *Autoformalization with large language models*. *Advances in Neural Information Processing Systems*, 35: 32353-32368 (2022).
[3] LU P; GONG R; JIANG S ET AL, *Inter-GPS: Interpretable geometry problem solving with formal language and symbolic reasoning*. *arXiv preprint* arXiv:2105.04165 (2021).
[4] HUANG D; SHI S; LIN C-Y; YIN J, *Learning fine-grained expressions to solve math word problems*. *Proceedings of the 2017 conference on empirical methods in natural language processing*, 805-814 (2017).