

Machine Learning for Automated Theorem Proving: an ML-side perspective

Boris Shminke¹

Expleo, Toulouse, France
`boris.shminke@expleogroup.com`

Introduction In my talk, I would like to give my perspective on applying machine learning (ML) to automated theorem proving (ATP). I will highlight several points which I found hindering the progress in known applications of ML to ATP, namely:

- importance of a benchmark for the development of ML models guiding proof search in ATP
- known problems and solutions in the task of data representation for ML models used to guide ATP
- comments and proposals of wider adoption of open source software development best practices to maintaining provers code bases (particularly in the case when ML guidance is involved)

I believe that a perspective from a person who applied ML to different business tasks (including banking and entertainment) and tried to guide ATP with ML might initiate currently dormant discussions and inspire other researchers to undertake new projects to the overall progress of ATP.

Benchmarking Current progress in computer vision and natural language processing tasks would not be possible without openly available and generally accepted benchmarks helping ML researchers to focus on building statistical models instead of specific business needs (that often are not limited to any ML in principle). Quite the contrary, in other fields like recommender systems [2] and graph neural networks [3], the lack or arguable quality of such benchmarks can lead to worrying signs of scientific stagnation. I will try demonstrating how ML in ATP is more similar to the latter and discuss possible solutions (based vaguely on existing solutions from other fields).

Representation Transformers recently revolutionised natural language processing like convolutional neural networks did for computer vision in previous years. In other domains (e.g. graph neural networks [1]), there is still no universally accepted faithful data representation. I will argue that a similar situation is characteristic of formal language processing, including but not limited to ATP. I will also briefly overview existing approaches in ATP and their known weaknesses.

Open Source Software Development The focus of research on publishing papers rather than writing code is a well-known problem [4], often limiting funding possibilities for core parts of many research projects. I will highlight several pain points I encountered in automated provers' development and propose possible ways to mitigate them in future.

References

- [1] Ijeoma Amuche Chikwendu, Xiaoling Zhang, Isaac Osei Agyemang, Isaac Adjei-Mensah, Ukwoma Chiagoziem Chima, and Chukwuebuka Joseph Ejayi. A comprehensive survey on deep graph representation learning methods. *J. Artif. Int. Res.*, 78, jan 2024.
- [2] Maurizio Ferrari Dacrema, Paolo Cremonesi, and D. Jannach. Are we really making much progress? a worrying analysis of recent neural recommendation approaches. *Proceedings of the 13th ACM Conference on Recommender Systems*, 2019.
- [3] Qingsong Lv, Ming Ding, Qiang Liu, Yuxiang Chen, Wenzheng Feng, Siming He, Chang Zhou, Jianguo Jiang, Yuxiao Dong, and Jie Tang. Are we really making much progress? revisiting, benchmarking and refining heterogeneous graph neural networks. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, KDD '21, page 1150–1160, New York, NY, USA, 2021. Association for Computing Machinery.
- [4] Chris Woolston. Why science needs more research software engineers. *Nature*, 2022.