First Datasets and Experiments with Neural Conjecturing

Josef Urban

Czech Institute of Informatics, Robotics and Cybernetics, Prague, Czech Republic

Abstract. We describe several datasets and first experiments with creating conjectures by neural methods. The datasets are based on the Mizar Mathematical Library processed in several forms and the problems extracted from it by the MPTP system and proved by the E prover using the ENIGMA guidance. The conjecturing experiments use the Transformer architecture and in particular its GPT-2 implementation.

1 Introduction and Related Work

Automated creation of suitable conjectures is one of the hard problems in automated reasoning over large mathematical corpora. This includes tasks such as (i) conjecturing suitable intermediate lemmas (cuts) when proving a harder conjecture, and (ii) unrestricted creation of interesting conjectures based on the previous theory (i.e., theory exploration). Starting with Lenat's AM [10], several systems such as the more specialized Graffitti by Fajtlowicz [4], and Colton's HR [3] have been developed, typically using heuristics for theory exploration or limited brute-force enumeration, controlled e.g. by the type system [7].

The motivation for our work are the experiments of Karpathy[1] with recurrent neural networks. One of his experiments included the Stacks project, generating LaTeX-style pseudo-mathematics that looked quite credible to non-experts. We have repeated these experiments over the Mizar library using Karpathy's neural architecture in 2016, but the results did not seem convincing. The neural methods have however improved since, coming up with stronger methods and systems such as attention, transformer and GPT-2 [11]. The experiments described here started just by testing GPT-2 on the Mizar library, gradually producing several more datasets and experiments.

Related work includes our research on the informal-to-formal grammar-based and neural translation [9,8,16,15]. There we have found that PCFGs and RNNs with attention work well on some informal-to-formal datasets, can learn analogies from the data, and can be used to produce multiple formal outputs of which some are new provable conjectures. In [15] we use this together with type checking to set up a data-augmentation loop between the neural learner and the type-checker. Such learning-reasoning loops are also planned for the datasets presented here. Similar, but more conjecture-oriented experiments are done in [6] and by Chvalovsky[2]. Gauthier has been working on term synthesis using Monte-Carlo Tree Search and reinforcement learning with semantic feedback [1,5].

[1] `http://karpathy.github.io/2015/05/21/rnn-effectiveness/`

[2] `http://aitp-conference.org/2019/abstract/AITP_2019_paper_27.pdf`, `http://aitp-conference.org/2020/abstract/paper_21.pdf`

2 Datasets

The datasets for neural conjecturing are available from our web page[3]. We have so far experimented with the following data:

1. All Mizar articles (MML version 1147), stripped of comments and concatenated together[4]. This is 78M of uncompressed text.
2. Text version of the HTML export [13] of the MML articles[5]. This unpacks to 156MB. It additionally contains disambiguation features such as full types of variables, full names of theorems and the thesis is printed after every natural deduction step. This seems useful for neural conjecturing because the context is repeated more often.
3. Tokenized TPTP proofs[6] of 28271 Mizar theorems translated by the MPTP system [14]. The proofs are produced by the E prover [12] equipped with the most recent ENIGMA guidance [2]. This unpacks to 658MB.
4. A subselection of the used Mizar premises from the 28271 proofs printed in prefix notation[7]. These files always start with the conjecture, and the premises are printed in the order in which E used them in its proof. This unpacks to 53MB.

Below we show short examples of the four kinds of data, all for the theorem `ZMODUL01:103`:

```
theorem
  for W being strict Submodule of V holds W /\ W = W
  proof
    let W be strict Submodule of V;
    the carrier of W = (the carrier of W) /\ (the carrier of W);
    hence thesis by Def15;
  end;

theorem :: ZMODUL01:103
for V being Z_Module
for W being strict Submodule of V holds W /\ W = W
proof
let V be Z_Module; ::_thesis: for W being strict Submodule of V holds W /\ W = W
let W be strict Submodule of V; ::_thesis: W /\ W = W
 the carrier of W = the carrier of W /\ the carrier of W ;
hence  W /\ W = W by Def15; ::_thesis: verum
end;

fof ( d15_zmodul01 , axiom , ! [ X1 ] : ( ( ( ( ( ( ( ( ( ( ~ ( v2_struct_0 ( X1 ) ) & ...
fof ( idempotence_k3_xboole_0 , axiom , ! [ X1 , X2 ] : k3_xboole_0 ( X1 , X1 ) = X1 ...
fof ( t103_zmodul01 , conjecture , ! [ X1 ] : ( ( ( ( ( ( ( ( ( ( ~ ( v2_struct_0 ( X1 ) ) ...
fof ( c_0_3 , plain , ! [ X118 , X119 , X120 , X121 ] : ( ( X121 ! = k7_zmodul01 ( X118 , ...
cnf ( c_0_6 , plain , ( X1 = k7_zmodul01 ( X4 , X2 , X3 ) | v2_struct_0 ( X4 ) | ...

c! b0  c=> c& c~ cv2_struct_0 b0 c& cv13_algstr_0 b0 c& cv2_rlvect_1 b0 c& cv3_rlvect_1 ...
c! b0  c=> c& c~ cv2_struct_0 b0 c& cv13_algstr_0 b0 c& cv2_rlvect_1 b0 c& cv3_rlvect_1 ...
c! b0   c! b1  c= ck3_xboole_0 b0 b0 b0
```

[3] http://grid01.ciirc.cvut.cz/~mptp/nn_conj20/

[4] http://grid01.ciirc.cvut.cz/~mptp/nn_conj20/datasets/mmlall.txt2

[5] http://grid01.ciirc.cvut.cz/~mptp/nn_conj20/datasets/html2.tar.gz

[6] http://grid01.ciirc.cvut.cz/~mptp/nn_conj20/datasets/prf2.tar.gz

[7] http://grid01.ciirc.cvut.cz/~mptp/nn_conj20/datasets/prf7.tar.gz

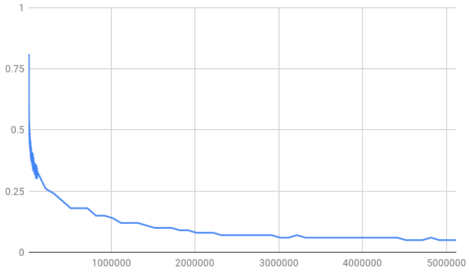

Fig. 1. Dataset 2 training and loss.

3 Experiments

The basic experiment for each dataset consists of training the smallest (117M parameters) version of GPT-2 on a NVIDIA GeForce GTX 1080 GPU with 12GB RAM, producing random unconditioned samples during the training. The produced samples and the most recent trained models are available from our web page[8]. The published models can be used for conditional and unconditional generation of Mizar-like texts, proofs and premise completion. The samples contain megabytes of examples of what can be generated and how the generated texts improve during the training. The training on the third dataset was stopped early. The large number of redundant tokens such as brackets and commas led us to produce the fourth dataset that uses the punctuation-free prefix notation and much shorter summary of the E proof (just the premises in their order). The training for datasets 1, 2 and 4 has been running for ca three weeks, and the performance is still improving. See Figure 1 for a sample training and loss on dataset 2. There are many interesting conjectures generated during the unconditioned sampling. The trained models can be directly used by Mizar users for autocompletion of their texts. Some examples compared to real theorems are shown below. More semantic evaluation on the textual datasets (1 and 2) could be done in various ways. We imagine that a proper Mizar environment will have to be guessed, some assumptions may be automatically added, etc.

```
# real MML theorem
theorem :: YELLOW10:61
for S, T being non empty up-complete Poset
for X being Subset of S
for Y being Subset of T st X is property(S) & Y is property(S) holds
[:X,Y:] is property(S)

# generated similar "theorem"
theorem :: WAYBEL30:56
for R being complete Lattice
for X being Subset of [:R,R:] st X is upper & X is property(S) holds
[:X,X:] is property(S)

# real MML theorem
theorem :: SEQ_4:47
for X being non empty real-membered set
```

[8] http://grid01.ciirc.cvut.cz/~mptp/nn_conj20/samples/, http://grid01.ciirc.cvut.cz/~mptp/nn_conj20/models/

```
for Y being real-membered set st X c= Y & Y is bounded_below holds
lower_bound Y <= lower_bound X

# generated similar "theorem"
theorem Th33: :: SEQ_4:33
for X being real-membered set
for Y being real-membered set st Y is bounded_below & X c= Y holds
lower_bound Y <= lower_bound X

# generated "theorem" with "proof"
theorem Th23: :: STIRL2_1:23
for X, Y being finite set st not X is empty & X c= Y & card X = card Y holds X = Y
proof
let X, Y be finite set ; ::_thesis: ( not X is empty & X c= Y & card X = card Y implies X = Y )
assume that
A1: not X is empty and A2: X c= Y and A3: card X = card Y ; ::_thesis: X = Y
 card (Y \ X) = (card Y) - (card X) by A1, A3, CARD_2:44;
then A4: card (Y \ X) = ((card Y) - 1) - (card X) by CARD_1:30;
 X = Y \ X by A2, A3, Th22;
hence  X = Y by A4, XBOOLE_0:def_10; ::_thesis: verum
end;
```

3.1 Conditioned completion

We have also done two experiments with proof and formula completion. The data and results for them are available from our web page[9]. In the first experiment, we use a model trained on Dataset 4 (premises), and ask the model to auto-complete 369 theorems from the CARD series of Mizar. For each conjecture we produce 10 premise selections using beam search, and we use different temperatures and beam search parameters. An interesting phenomenon is that with low temperatures, practically all conjectured premises are known Mizar theorems. I.e., the task reduces to standard premise selection. With higher temperatures, GPT-2 starts producing premises (lemmas) that are not among the existing Mizar theorems, but are still well-typed. Even higher temperatures lead to non-well-typed or even unparsable lemmas. One of the next steps could be a suitable ATP evaluation. I.e., we could try to see if the lemmas prove the conjecture, and if the new lemmas are provable from known MML theorems.

The second experiment was done over Dataset 2 and a set of 462 partial formulas from the CARD articles. The model trained on Dataset 2 is then (again using beam search) asked to auto-complete these formulas. Mizar users can also play with such autocompletion via a web server[10] using this model. For example,

```
for M, N being Cardinal holds
```

results in the following (selected) auto-completions (more are available online[11]):

```
( M = N iff M,N are_equipotent )
( M = N iff not 0 in M ) by ORDINAL3:8;
M *` N = N *` M
( M in N iff not M c= N )
the_rank_of M = the_rank_of N by Th77;
nextcard (Sum M) = M *` N
```

[9] http://grid01.ciirc.cvut.cz/~mptp/nn_conj20/samples/premises/, http://grid01.ciirc.cvut.cz/~mptp/nn_conj20/samples/html2/

[10] http://grid01.ciirc.cvut.cz:8000/

[11] http://grid01.ciirc.cvut.cz/~mptp/nn_conj20/samples/html2/00cardmizout1_t1

References

1. Chad E. Brown and Thibault Gauthier. Self-learned formula synthesis in set theory. *CoRR*, abs/1912.01525, 2019.
2. Karel Chvalovský, Jan Jakubuv, Martin Suda, and Josef Urban. ENIGMA-NG: efficient neural and gradient-boosted inference guidance for E. In *CADE 27*, pages 197–215, 2019.
3. Simon Colton. *Automated Theory Formation in Pure Mathematics*. Distinguished Dissertations. Springer London, 2012.
4. Siemion Fajtlowicz. On conjectures of Graffiti. *Annals of Discrete Mathematics*, 72(1–3):113–118, 1988.
5. Thibault Gauthier. Deep reinforcement learning in HOL4. *CoRR*, abs/1910.11797, 2019.
6. Thibault Gauthier, Cezary Kaliszyk, and Josef Urban. Initial experiments with statistical conjecturing over large formal corpora. In *CICM'16 WiP Proceedings*, pages 219–228, 2016.
7. Moa Johansson, Dan Rosén, Nicholas Smallbone, and Koen Claessen. Hipster: Integrating theory exploration in a proof assistant. In *CICM 2014*, pages 108–122, 2014.
8. Cezary Kaliszyk, Josef Urban, and Jirí Vyskocil. Automating formalization by statistical and semantic parsing of mathematics. In Mauricio Ayala-Rincón and César A. Muñoz, editors, *Interactive Theorem Proving - 8th International Conference, ITP 2017, Brasília, Brazil, September 26-29, 2017, Proceedings*, volume 10499 of *Lecture Notes in Computer Science*, pages 12–27. Springer, 2017.
9. Cezary Kaliszyk, Josef Urban, and Jiří Vyskočil. Learning to parse on aligned corpora (rough diamond). In *ITP 2015*, pages 227–233, 2015.
10. Douglas Bruce Lenat. *AM: An Artificial Intelligence Approach to Discovery in Mathematics as Heuristic Search*. PhD thesis, Stanford, 1976.
11. Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8):9, 2019.
12. Stephan Schulz. System description: E 1.8. In Kenneth L. McMillan, Aart Middeldorp, and Andrei Voronkov, editors, *LPAR*, volume 8312 of *LNCS*, pages 735–743. Springer, 2013.
13. Josef Urban. XML-izing Mizar: Making semantic processing and presentation of MML easy. In Michael Kohlhase, editor, *MKM*, volume 3863 of *LNCS*, pages 346–360. Springer, 2005.
14. Josef Urban. MPTP 0.2: Design, implementation, and initial experiments. *J. Autom. Reasoning*, 37(1-2):21–43, 2006.
15. Qingxiang Wang, Chad E. Brown, Cezary Kaliszyk, and Josef Urban. Exploration of neural machine translation in autoformalization of mathematics in mizar. In Jasmin Blanchette and Catalin Hritcu, editors, *Proceedings of the 9th ACM SIGPLAN International Conference on Certified Programs and Proofs, CPP 2020, New Orleans, LA, USA, January 20-21, 2020*, pages 85–98. ACM, 2020.
16. Qingxiang Wang, Cezary Kaliszyk, and Josef Urban. First experiments with neural translation of informal to formal mathematics. In Florian Rabe, William M. Farmer, Grant O. Passmore, and Abdou Youssef, editors, *Intelligent Computer Mathematics - 11th International Conference, CICM 2018, Hagenberg, Austria, August 13-17, 2018, Proceedings*, volume 11006 of *Lecture Notes in Computer Science*, pages 255–270. Springer, 2018.