First Neural Conjecturing Datasets and Experiments

Josef Urban Jan Jakubův

Czech Technical University in Prague

CICM 2020, July 29th, 2020

Outline

Conjecturing and Neural Language models

Datasets and Training

Evaluation

Conjecturing in mathematics

- Targeted: generate intermediate lemmas (cuts) for a harder conjecture
- Unrestricted (theory exploration):
- Creation of interesting conjectures based on the previous theory
- One of the most interesting activities mathematicians do (how?)
- Higher-level AI/reasoning task - can we learn it?
- If so, we have solved math:
- ... just (recursively) divide Fermat into many subtasks ...
- ... and conquer (I mean: hammer) them away

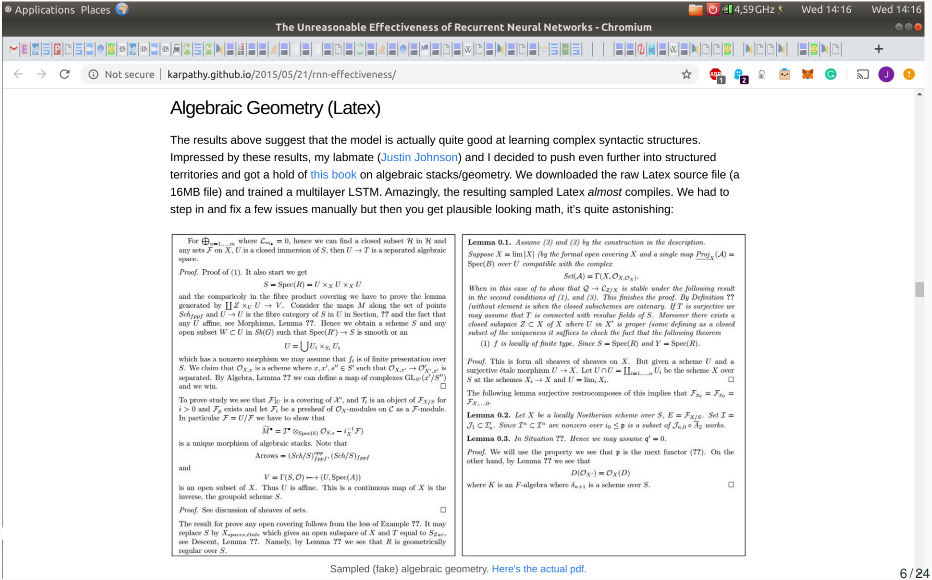
A bit of conjecturing history

- The topic goes back at least to Lenat (AM) and Fajtlowicz (Graffiti)
- Combined with automated theorem proving by Colton et al. in early 2000s (HR)
- Theory exploration for Isabelle by Johansson et al (Hipster)
- Several learning-based/neural approaches by our groups since 2015:
- Based mainly on learning analogies and informalization followed by probabilistic/neural disambiguation ...
- ... Gauthier, Kaliszyk, Chvalovsky, Piotrowski, Goertzel, Wang, Brown, JU

Neural language models - RNNs, Transformers, GPT

- RNNs (recurrent neural nets) for machine translation (Mikolov 2010/12)
- Karpathy'15 - RNN experiments with generating fake Math over Stacks
- Greatly improved on linguistic tasks by a mechanism called attention:
- Learn to "attend to" a certain part of the input
- Evolved into Transformer (2017) - multiple attention layers
- GPT (-2,3) - large language models based on Transformer
- Millions/billions of parameters
- Capable of generating quite credible texts
- Let's try to use them for formal-math tasks and combine with ATP!

Karpathy’s RNN Trained on Stacks



Algebraic Geometry (Latex)

The results above suggest that the model is actually quite good at learning complex syntactic structures. Impressed by these results, my labmate (Justin Johnson) and I decided to push even further into structured territories and got a hold of this book on algebraic stacks/geometry. We downloaded the raw Latex source file (a 16MB file) and trained a multilayer LSTM. Amazingly, the resulting sampled Latex *almost* compiles. We had to step in and fix a few issues manually but then you get plausible looking math, it’s quite astonishing:

For $\bigoplus_{n=1,\ldots,m}$ where $\mathcal{L}_{m_\bullet} = 0$, hence we can find a closed subset $\mathcal{H}$ in $\mathcal{H}$ and any sets $\mathcal{F}$ on X, U is a closed immersion of S, then $U \to T$ is a separated algebraic space.

Proof. Proof of (1). It also start we get

$$S = \mathrm{Spec}(R) = U \times_X U \times_X U$$

and the comparicoly in the fibre product covering we have to prove the lemma generated by $\coprod Z \times_U U \to V$. Consider the maps M along the set of points Sch_{fppf} and $U \to U$ is the fibre category of S in U in Section, **??** and the fact that any U affine, see Morphisms, Lemma **??**. Hence we obtain a scheme S and any open subset $W \subset U$ in $Sh(G)$ such that $\mathrm{Spec}(R') \to S$ is smooth or an

$$U = \bigcup U_i \times_{S_i} U_i$$

which has a nonzero morphism we may assume that f_i is of finite presentation over S. We claim that $\mathcal{O}_{X,x}$ is a scheme where $x, x', s'' \in S'$ such that $\mathcal{O}_{X,x'} \to \mathcal{O}'_{X',x'}$ is separated. By Algebra, Lemma **??** we can define a map of complexes $\mathrm{GL}_{S'}(x'/S'')$ and we win. □

To prove study we see that $\mathcal{F}|_U$ is a covering of $\mathcal{X}'$, and $\mathcal{T}_i$ is an object of $\mathcal{F}_{X/S}$ for $i > 0$ and $\mathcal{F}_p$ exists and let $\mathcal{F}_i$ be a presheaf of $\mathcal{O}_X$-modules on $\mathcal{C}$ as a $\mathcal{F}$-module. In particular $\mathcal{F} = U/\mathcal{F}$ we have to show that

$$\widetilde{M}^\bullet = \mathcal{I}^\bullet \otimes_{\mathrm{Spec}(k)} \mathcal{O}_{S,s} - i_X^{-1}\mathcal{F})$$

is a unique morphism of algebraic stacks. Note that

$$\mathrm{Arrows} = (Sch/S)^{opp}_{fppf}, (Sch/S)_{fppf}$$

and

$$V = \Gamma(S, \mathcal{O}) \longmapsto (U, \mathrm{Spec}(A))$$

is an open subset of X. Thus U is affine. This is a continuous map of X is the inverse, the groupoid scheme S.

Proof. See discussion of sheaves of sets. □

The result for prove any open covering follows from the less of Example **??**. It may replace S by $X_{spaces,\acute{e}tale}$ which gives an open subspace of X and T equal to S_{Zar}, see Descent, Lemma **??**. Namely, by Lemma **??** we see that R is geometrically regular over S.

Lemma 0.1. *Assume (3) and (3) by the construction in the description.*

Suppose $X = \lim |X|$ (by the formal open covering X and a single map $\underline{Proj}_X(\mathcal{A}) =$ $\mathrm{Spec}(B)$ over U compatible with the complex

$$Set(\mathcal{A}) = \Gamma(X, \mathcal{O}_{X,\mathcal{O}_X}).$$

*When in this case of to show that $\mathcal{Q} \to \mathcal{C}_{Z/X}$ is stable under the following result in the second conditions of (1), and (3). This finishes the proof. By Definition **??** (without element is when the closed subschemes are catenary. If T is surjective we may assume that T is connected with residue fields of S. Moreover there exists a closed subspace $Z \subset X$ of X where U in X' is proper (some defining as a closed subset of the uniqueness it suffices to check the fact that the following theorem*

(1) *f is locally of finite type. Since $S = \mathrm{Spec}(R)$ and $Y = \mathrm{Spec}(R)$.*

Proof. This is form all sheaves of sheaves on X. But given a scheme U and a surjective étale morphism $U \to X$. Let $U \cap U = \coprod_{i=1,\ldots,n} U_i$ be the scheme X over S at the schemes $X_i \to X$ and $U = \lim_i X_i$. □

The following lemma surjective restrocomposes of this implies that $\mathcal{F}_{x_0} = \mathcal{F}_{x_0} = \mathcal{F}_{\mathcal{X},\ldots,0}$.

Lemma 0.2. *Let X be a locally Noetherian scheme over S, $E = \mathcal{F}_{X/S}$. Set $\mathcal{I} = \mathcal{J}_1 \subset \mathcal{I}'_n$. Since $\mathcal{I}^n \subset \mathcal{I}^n$ are nonzero over $i_0 \leq \mathfrak{p}$ is a subset of $\mathcal{J}_{n,0} \circ \overline{A}_2$ works.*

Lemma 0.3. *In Situation **??**. Hence we may assume $\mathfrak{q}' = 0$.*

Proof. We will use the property we see that $\mathfrak{p}$ is the mext functor (**??**). On the other hand, by Lemma **??** we see that

$$D(\mathcal{O}_{X'}) = \mathcal{O}_X(D)$$

where K is an F-algebra where δ_{n+1} is a scheme over S. □

Sampled (fake) algebraic geometry. Here's the actual pdf.

Full Mizar-based datasets for the GPT-2 Training

1. `http://grid01.ciirc.cvut.cz/~mptp/nn_conj20/`
2. All Mizar articles, stripped of comments and concatenated together (78M)
3. Articles with added context/disambiguation (156M) (types, names, thesis)
4. TPTP proofs of 28271 Mizar/MPTP theorems by E/ENIGMA (658M)
5. Just the conjecture and premises needed for the 28271 proofs printed in prefix notation

The same example in the four datasets

```
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 ( X
fof ( c_0_3 , plain , ! [ X118 , X119 , X120 , X121 ] : ( ( X121 ! = k7_zmodul01 ( X1
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
c! b0  c=> c& c~ cv2_struct_0 b0 c& cv13_algstr_0 b0 c& cv2_rlvect_1 b0 c& cv3_rlvect
c! b0   c! b1  c= ck3_xboole_0 b0 b0 b0
```

Training GPT-2

- Train GPT-2 for several weeks on the datasets
- Save the models for later evaluation
- Print unconditioned samples produced during the training
- Megabytes of conjectures and "proofs" thus available for evaluation
- Addictive experience - don't look at the samples - too much fun!
- The GPT-2 (linguistic) loss still decreasing after several weeks

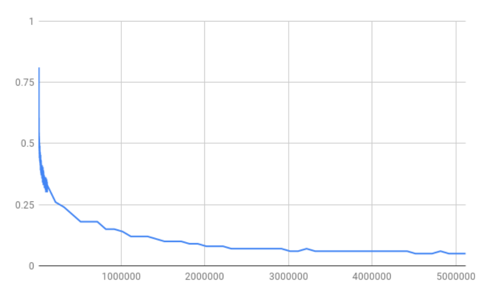


Figure: Dataset 2 training and loss.

Examples of similar theorems generated

```
# 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 statement (nontrivial instantiation)
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
for Y being real-membered set st X c= Y & Y is bounded_below holds
lower_bound Y <= lower_bound X

# generated similar statement (non emptiness omitted)
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
```

Can you find the flaw(s) in this fake GPT-2 proof?

Applications Places — emacs@dell — Wed 15:02

File Edit Options Buffers Tools Index Mizar Hide/Show Help

```
:: 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;
```

-:--- **card_tst.miz** 99% L2131 (Mizar Errors:13 hs Undo-Tree)

Figure: Fake full declarative GPT-2 “proof” - typechecks!

Conditioned Completion - small premise selection eval

- Use a trained model to autocomplete a particular formula/term/proof
- Done first on dataset 4 for a smaller selection of theorems about cardinals
- 369 theorems from the CARD series of Mizar: generate premises
- Done later in a more complete way with full ATP eval
- Higher temperatures: new formulas generated, can contain new symbols
- Lower temperatures: new formulas but parsable/typable
- Even lower temperatures: only existing premises generated
- Smooth transition from premise selection to conjecturing to symbol generation!

Conditioned Completion - original Mizar

- Used a model trained on Dataset 2
- Evaluated on a set of 462 partial formulas from the CARD articles
- Have fun online! - `http://grid01.ciirc.cvut.cz:5500/`

```
for M, N being Cardinal holds

GPT-completes to:

( 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
```

Mizar autocompletion server in action

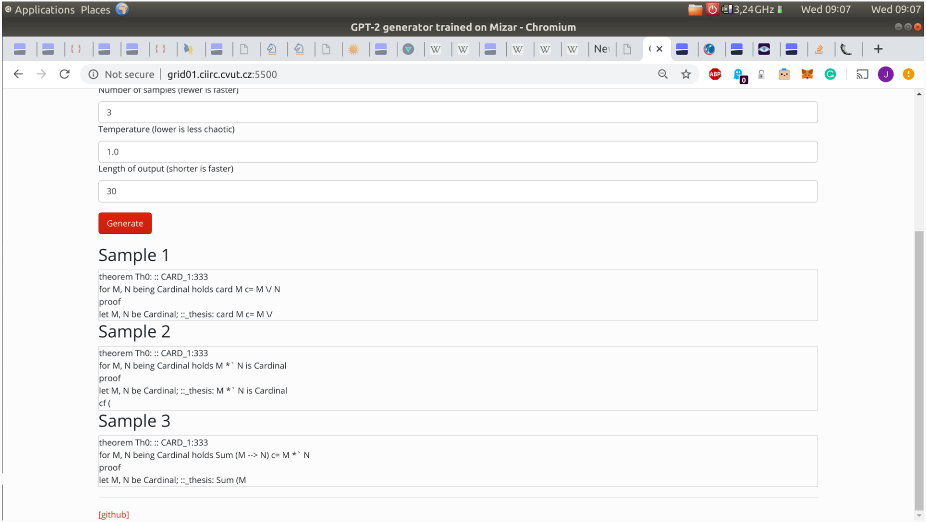
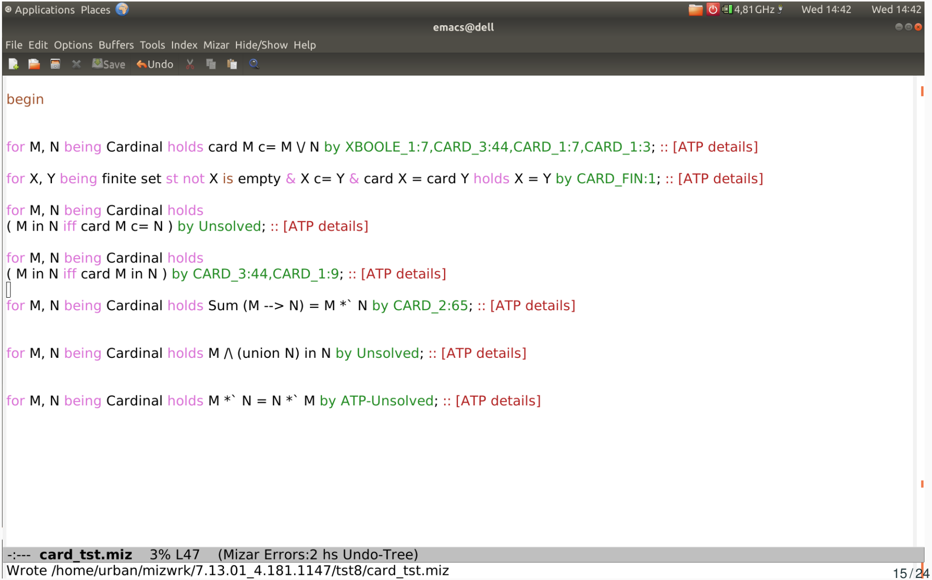


Figure: MGG - Mizar Gibberish Generator.

Proving the conditioned completions - MizAR hammer



Initial ATP Evaluation - part 1

- Uses Dataset 4 - ATP-ready conjectures and premises
- And a GPT-2 model M trained on the 28k examples
- M evaluated on 31792 Mizar theorems of which 6639 are not in the training set
- For each we produce 12 sets of premise predictions
- Yields 381432 predictions, deduplicated to 193320
- For 108564 no new conjectures - works as a premise selector
- 86899 of them CounterSatisfiable - linguistic loss differs from premise-selection loss!
- 11866 provable in 6s by E - proofs of 8105 theorems
- The premises also do not obey the MML chronological order
- Some new proofs however obtained - see the paper

Initial ATP Evaluation - part 2

- 44524 problems use at least one newly proposed premise (cut)
- To partially satisfy the chronology, we remove the theorem itself if it appears
- For 1515 problems a proof is found using the cut
- We use this as the first interestingness filter for the cuts
- The cuts may be however hard to prove.
- Kinyon and Stanovsky (algebraists) confirmed that this cut is valid:

```
theorem Th10: :: GROUPP_1:10
for G being finite Group for N being normal Subgroup of G st
N is Subgroup of center G & G ./. N is cyclic holds G is commutative

The generalization that avoids finiteness:

for G being Group for N being normal Subgroup of G st
N is Subgroup of center G & G ./. N is cyclic holds G is commutative
```

Gibberish Generator Provoking Algebraists

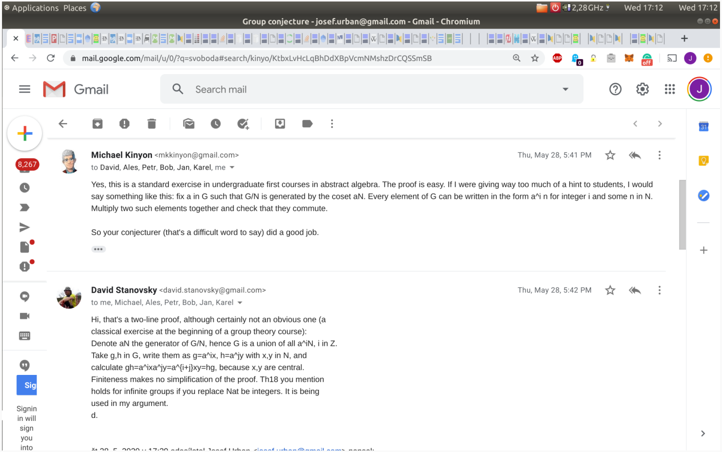


Figure: First successes in making mathematicians comment on AI.

More cuts

- In total 33100 in this experiment
- Ca 9k proved by trained ENIGMA
- Some are clearly false, yet quite natural to ask:

```
theorem :: SINCOS10:17
sec is increasing on [0, pi/2)

leads to conjecturing the following:

Every differentiable function is increasing.
```

Conclusion and Future Work

- Neural conjecturing is good fun!
- The attention-based architectures can at least memorize ...
- ... and to some extent consistently analogize ...
- ... which sometimes also means generalize and instantiate
- This seems to be just the beginning ...
- ... we can train in many other ways
- ... do the learning/proving loop
- ... redefine the loss for AI/TP tasks
- ... try more targeted architectures
- ... etc ...

Thanks and Advertisement

- Thanks for your attention! Questions?
- AITP – Artificial Intelligence and Theorem Proving
- March 22–27 ==> September, 2020, Aussois, France, `aitp-conference.org`
- ATP/ITP/Math vs AI/Machine-Learning people, Computational linguists
- Discussion-oriented and experimental
- Grown to 80 people in 2019

Some References - Conjecturing

- Douglas Bruce Lenat. *AM: An Artificial Intelligence Approach to Discovery in Mathematics as Heuristic Search*. PhD thesis, Stanford, 1976.
- Siemion Fajtlowicz. On conjectures of Graffiti. *Annals of Discrete Mathematics*, 72(1–3):113–118, 1988.
- Simon Colton. *Automated Theory Formation in Pure Mathematics*. Distinguished Dissertations. Springer London, 2012.
- Moa Johansson, Dan Rosén, Nicholas Smallbone, and Koen Claessen. Hipster: Integrating theory exploration in a proof assistant. In *CICM 2014*, pages 108–122, 2014.
- 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.
- Thibault Gauthier, Cezary Kaliszyk: Sharing HOL4 and HOL Light Proof Knowledge. LPAR 2015: 372-386
- Thibault Gauthier. Deep reinforcement learning in HOL4. *CoRR*, abs/1910.11797, 2019.
- Chad E. Brown and Thibault Gauthier. Self-learned formula synthesis in set theory. *CoRR*, abs/1912.01525, 2019.
- Bartosz Piotrowski, Josef Urban, Chad E. Brown, Cezary Kaliszyk: Can Neural Networks Learn Symbolic Rewriting? AITP 2019, CoRR abs/1911.04873 (2019)
- Zarathustra Goertzel and Josef Urban. Usefulness of Lemmas via Graph Neural Networks (Extende Abstract). AITP 2019.
- Karel Chvalovský, Thibault Gauthier and Josef Urban: First Experiments with Data Driven Conjecturing (Extended Abstract). AITP 2019.
- Thibault Gauthier: Deep Reinforcement Learning for Synthesizing Functions in Higher-Order Logic. LPAR 2020: 230-248
- Bartosz Piotrowski, Josef Urban: Guiding Inferences in Connection Tableau by Recurrent Neural Networks. CICM 2020: 309-314

References on PCFG and Neural Autoformalization - with some Conjecturing Applications

- Cezary Kaliszyk, Josef Urban, Jirí Vyskocil: Learning to Parse on Aligned Corpora (Rough Diamond). ITP 2015: 227-233
- Cezary Kaliszyk, Josef Urban, Jirí Vyskocil, Herman Geuvers: Developing Corpus-Based Translation Methods between Informal and Formal Mathematics: Project Description. CICM 2014: 435-439
- C. Kaliszyk, J. Urban, J. Vyskocil: Automating Formalization by Statistical and Semantic Parsing of Mathematics. ITP 2017: 12-27
- Cezary Kaliszyk, Josef Urban, Jirí Vyskocil: System Description: Statistical Parsing of Informalized Mizar Formulas. SYNASC 2017: 169-172
- Q. Wang, C. Kaliszyk, J. Urban: First Experiments with Neural Translation of Informal to Formal Mathematics. CICM 2018: 255-270
- Qingxiang Wang, Chad E. Brown, Cezary Kaliszyk, Josef Urban: Exploration of neural machine translation in autoformalization of mathematics in Mizar. CPP 2020: 85-98

Some References on the Corpora and Methods - Mizar (hammers), E/ENIGMA

- 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.
- Josef Urban. MPTP 0.2: Design, implementation, and initial experiments. *J. Autom. Reasoning*, 37(1-2):21–43, 2006.
- Josef Urban, Piotr Rudnicki, Geoff Sutcliffe: ATP and Presentation Service for Mizar Formalizations. J. Autom. Reasoning 50(2): 229-241 (2013)
- Cezary Kaliszyk, Josef Urban: MizAR 40 for Mizar 40. J. Autom. Reasoning 55(3): 245-256 (2015)
- Stephan Schulz: System Description: E 1.8. LPAR 2013: 735-743
- Jan Jakubuv, Josef Urban: ENIGMA: Efficient Learning-Based Inference Guiding Machine. CICM 2017: 292-302
- Karel Chvalovský, Jan Jakubuv, Martin Suda, Josef Urban: ENIGMA-NG: Efficient Neural and Gradient-Boosted Inference Guidance for E. CADE 2019: 197-215
- Jan Jakubuv, Josef Urban: Hammering Mizar by Learning Clause Guidance. ITP 2019: 34:1-34:8
- Jan Jakubuv, Karel Chvalovský, Miroslav Olsák, Bartosz Piotrowski, Martin Suda, Josef Urban: ENIGMA Anonymous: Symbol-Independent Inference Guiding Machine. IJCAR (2) 2020: 448-463