

Autoformalization of Mathematics

Markus Wich

20.1.2021

- Automated theorem proving
- Interactive theorem proving/Proof assistants

Requirement

Mathematics need to be in a computer-understandable format
(Formalized language)

- Most Mathematics today are produced by mathematicians in papers
- Usually written in LaTeX (Informal language)
- Efforts being made to formalize parts of the mathematical corpus
- Examples for formalized libraries: Mizar, HOL Light, Isabelle
- Formalization by hand requires domain experts

⇒ Speed of formalization slower than increase of mathematics

⇒ Possible Solution: Automate the process (Autoformalization)

Previous Research on Autoformalization

- Wang (1954): Formalization of Mathematics
- QED Manifesto (1994): Vision for database of formalized mathematics
- Zinn (2004): Theory book parsing via domain discourse theory
- Kaliszyk et al. (2017): Parsing informal latex formulas with probabilistic context-free grammars

General consensus

Autoformalization is a very hard/impossible task

Autoformalization through Machine Translation

Idea

Treat autoformalization as a language translation problem

⇒ Allows the use of already existing machine translation tools

Definition

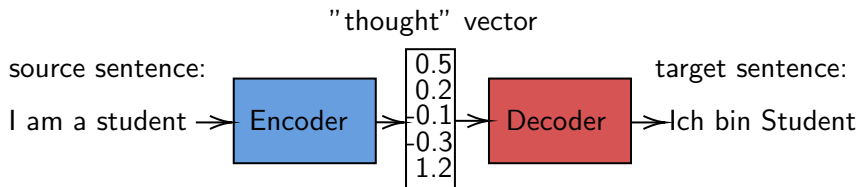
Machine Translation: Translating text (or speech) from one language to another

Definition

Neural Machine Translation (NMT): Machine translation by using neural networks

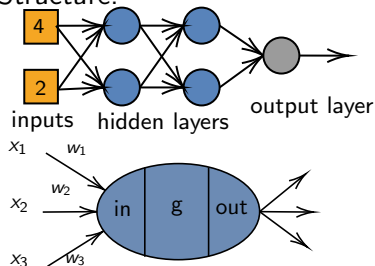
- Models
 - seq2seq model
 - Unsupervised NMT (UNMT)
 - cross-lingual model (XLM)
- Datasets
- Data augmentation mechanism

- Models
 - seq2seq model
 - Unsupervised NMT (UNMT)
 - cross-lingual model (XLM)
- Datasets
- Data augmentation mechanism



Short explanation of neural networks

Structure:

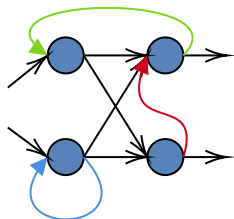


$$in = \sum_i^n x_i \cdot w_i \quad (1)$$

$$g(x) : \text{activation function} \quad (2)$$

$$out = g(in) \quad (3)$$

Recurrent neural network



- Direct feedback
- Indirect feedback
- Lateral feedback

Recurrent neural networks are particularly useful when dealing with sequences of data as input.

Long short-term memory cell

Problem

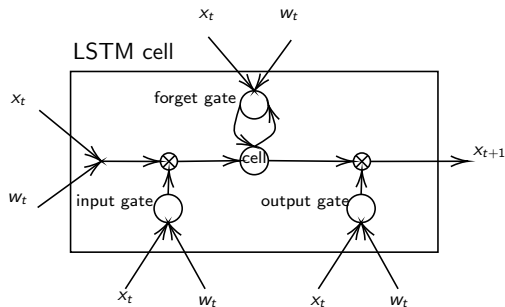
Problems when training large neural networks with backpropagation:

- Exploding gradients
- Vanishing gradients

Solution

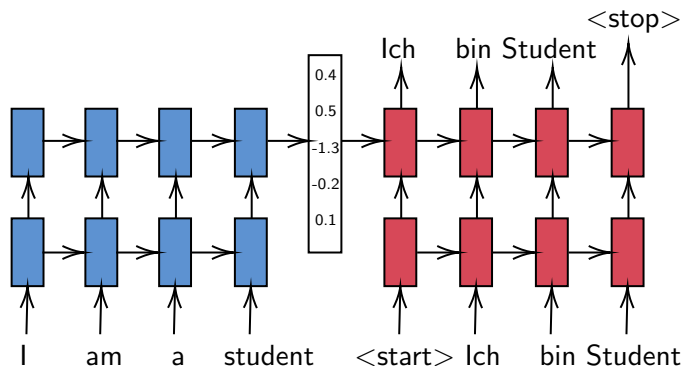
Long short-term memory cells

Long short-term memory cell

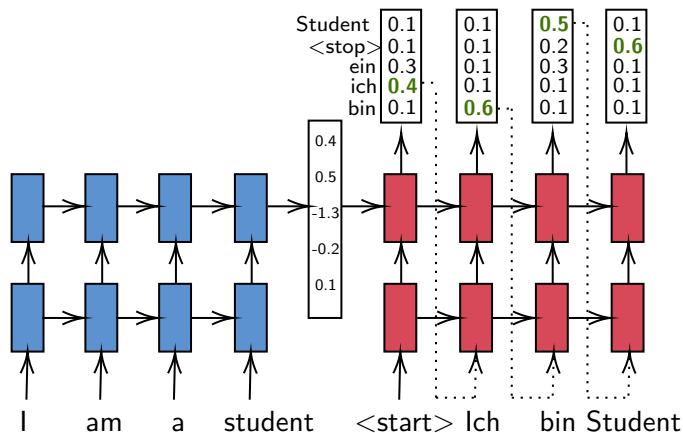


- input gate: decides how much a new value should influence the cell
- forget gate: decides how long a value should stay in the cell or how fast it should be forgotten
- output gate: decides how much the cell value should influence the next cell

seq2seq NMT Training



seq2seq NMT Inference



seq2seq NMT attention mechanism

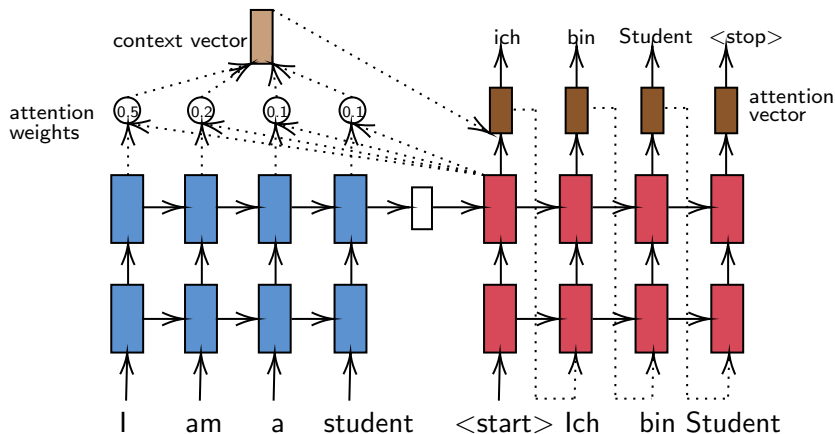
Problem

Thought vector not good enough when dealing with long sentences

Solution

Attention mechanism

seq2seq NMT attention mechanism



- Models
 - seq2seq model
 - Unsupervised NMT (UNMT)
 - cross-lingual model (XLM)
- Datasets
- Data augmentation mechanism

Unsupervised NMT model (UNMT)

Problem

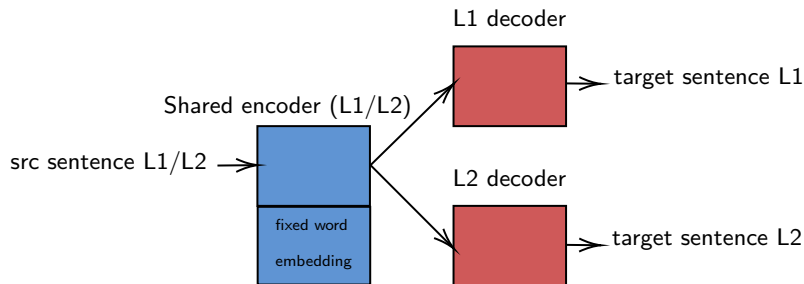
A sufficiently large corpus of aligned data is expensive

Idea

Do NMT as an unsupervised task

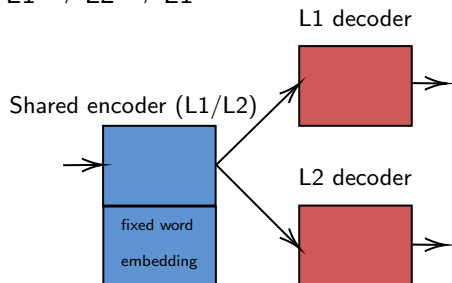
⇒ Turn the unsupervised problem into a series of supervised problems

Unsupervised NMT model (UNMT)



Unsupervised NMT model (UNMT)

$L1 \rightarrow L2 \rightarrow L1$



- Models
 - seq2seq model
 - Unsupervised NMT (UNMT)
 - cross-lingual model (XLM)
- Datasets
- Data augmentation mechanism

Cross-lingual pretrained model (XLM)

- Supports both supervised and unsupervised learning, but only unsupervised is used
- In the UNMT model the vector representation of word tokens is fixed
- Pretraining is used to obtain the word embedding

- Models
 - seq2seq model
 - Unsupervised NMT (UNMT)
 - cross-lingual model (XLM)
- Datasets
- Data augmentation mechanism

Dataset 1: Synthetic LaTeX - Mizar Dataset

Problem

A sufficiently large dataset of aligned LaTeX - Mizar is expensive

Temporary workaround

Use informalization to generate sentence pairs

Downside

Sentences are artificial and use a smaller vocabulary than a human would

⇒ Model can not be generalized for real data

dataset	src	target	src sentences	target sentences	aligned
synthetic	LaTeX	Mizar	1e6	1e6	1e6

Dataset 1 examples

Mizar	let T be RelStr ;
LaTeX	Let T be a relational structure .
L ^A T _E X	Let T be a relational structure .
Mizar	$a \text{ ast } (b \text{ ast } t) \leq b \text{ ast } t ;$
LaTeX	$\$ a \backslash \text{ast } (b \backslash \text{ast } t) \backslash \text{leq } b \backslash \text{ast } t \$.$
L ^A T _E X	$a * (b * t) \leq b * t.$

Dataset 2 and 3: ProofWiki - Mizar Datasets

- Collected sentences from proofwiki.org (MathJax) and the Mizar Math library
- For the topology choose only the sentences from the same topic realm

dataset	src	target	src sentences	target sentences	aligned
proofwiki full	LaTeX	Mizar	2e5	1e6	330
proofwiki topology	LaTeX	Mizar	30000	50000	330

Dataset 2 and 3 examples

- Mizar for T being non empty TopSpace for A being Subset of T st A is countable holds $A^0 = \{ \}$
- LaTeX Let $T = \left(\{S, \tau\} \right)$ be a topological space. Let A be a subset of S . Then if A is countable, then $A^0 = \text{\texttt{\textbackslash varnothing}}$.
- ℒ_ATeX Let $T = (\{S, \tau\})$ be a topological space. Let A be a subset of S . Then if A is countable, then $A^0 = \emptyset$.

Dataset 4: Mizar - TPTP

- Additional dataset to test the data augmentation mechanism
- This aligned Mizar - TPTP dataset can be generated with the Mizar toolchain

dataset	src	target	src sentences	target sentences	aligned
TPTP	Mizar	TPTP Prefix Format	54000	54000	54000

Dataset 4 example

Mizar for A holds A is doubleLoopStr & not A is empty implies B hold B is Scalar of
 A implies B is being_a_sqaare iff ex C st C is Scalar of A & $B = C^2$

Prefix c! b0 c=>__2 c&__2 cn16_algstr_0__1 b0 c~__1 cnv2_struct_0__1 b0 c! b1
 c=>__2 cnm4_vectsp_1__2 b1 b0 c<=>__2 cnv1_o_ring_1__1 b1 c? b2 c&__2
 cnm4_vectsp_1__2 b2 b0 cnr1_hidden__2 b1 cnk1_o_ring_1__1 b2

- Models
 - seq2seq model
 - Unsupervised NMT (UNMT)
 - cross-lingual model (XLM)
- Datasets
- Data augmentation mechanism

Augmentation with feed back loop

Definition Augmentation

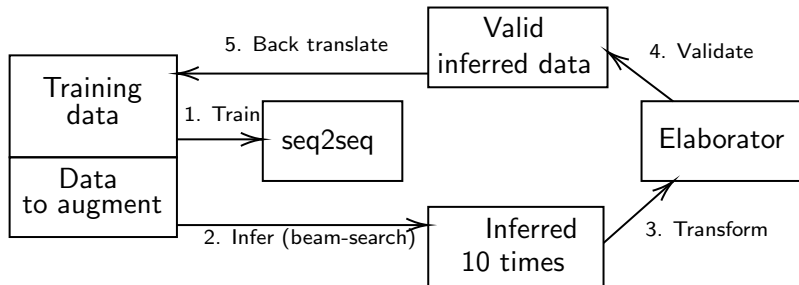
Data augmentation increases the amount of data by artificially creating new entries

Idea

Augment the data by generating new sentences with a feedback loop

- 1 Use beam search instead of greedy search during the inference of the seq2seq model
- 2 Check if some of the new target sentences can be translated back to the source sentence
- 3 Add these new sentence pair to the training data and retrain the model

Augmentation with feed back loop



Metrics:

- Bilingual evaluation understudy (BLEU) rate: Counting of n-grams
- Perplexity score: Counting of n-grams
- Levenshtein edit distance: Number of insertions, deletions and substitutions the result is off the correct sentence

Disclaimer

All these metrics only compare syntactic closeness!

dataset	src	target	src sentences	target sentences	aligned
synthetic	LaTeX	Mizar	1e6	1e6	1e6
proofwiki full	LaTeX	Mizar	2e5	1e6	330
proofwiki topology	LaTeX	Mizar	30000	50000	330

Synthetic latex - Mizar dataset:

2000 samples	seq2seq	UNMT	XLM
BLEU	70.9	27.14	43.61
Perplexity	1.58	3.00	2.91
Edit distance 0	65.2%	26.8%	34.1%
Edit distance ≤ 1	74.6%	34.4%	38.5%
Edit distance ≤ 2	81.5%	41.8%	42.1%
Edit distance ≤ 3	83.9%	46.3%	45.9%

Evaluation

dataset	src	target	src sentences	target sentences	aligned
synthetic	LaTeX	Mizar	1e6	1e6	1e6
proofwiki full	LaTeX	Mizar	2e5	1e6	330
proofwiki topology	LaTeX	Mizar	30000	50000	330

Proofwiki - Mizar datasets:

131 samples	UNMT		XLM	
	topology	full	topology	full
BLEU	4.03	1.55	7.87	6.07
Perplexity	11.57	10.73	33.01	39.70
Edit distance 0	0%	0%	0%	0%
Edit distance ≤ 1	0%	0%	0%	0%
Edit distance ≤ 2	0%	0%	0.76%	0%
Edit distance ≤ 3	9.92%	2.29%	6.11%	2.29%

Evaluation of augmentation

dataset	src	target	src sentences	target sentences	aligned
TPTP	Mizar	TPTP Prefix Format	54000	54000	54000

Mizar - TPTP:

2000 samples	Iter. 1	Iter. 2	Iter. 3	Iter. 4	Iter. 5
BLEU	42.3	85.5	88.7	88.0	87.4
Perplexity	2.68	1.27	1.14	1.18	1.17
Edit distance 0	2.05%	2.40%	2.00%	5.15%	4.45%
Edit distance ≤ 1	9.70%	13.4%	20.9%	21.6%	19.2%
Edit distance ≤ 2	22.3%	25.0%	38.7%	36.9	34.6%
Edit distance ≤ 3	32.6%	34.3%	49.3%	48.5%	45.4%

Closing remarks

- Autoformalization via NMT is possible to some degree
 - Created a working data augmentation mechanism
- Lack of sizeable dataset
 - No metric to check for semantic closeness

Conclusion

The experiments show that autoformalization via NMT is a promising research topic, but there are still some challenges to overcome before this can be applied to real world data