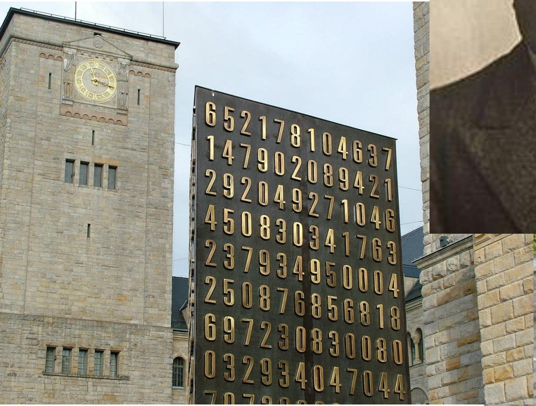
Marian: Homecoming

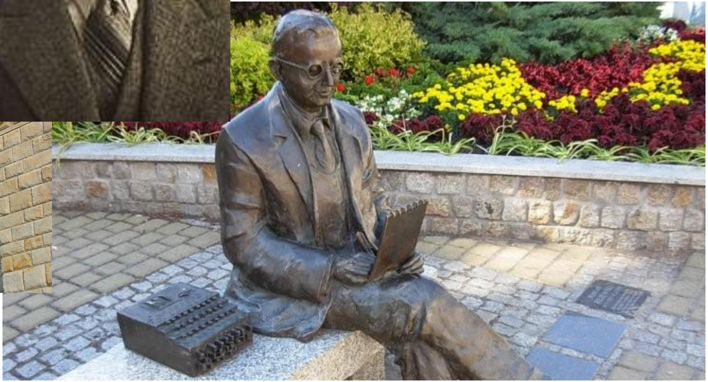
Marcin Junczys-Dowmunt

Microsoft

Translator

Marian Rejewski





A few words about Marian

- Portable C++ code with minimal dependencies (CUDA or MKL and still Boost);
- Single engine for training and decoding on GPU and CPU;
- Custom auto-diff engine with dynamic graphs (similar to DyNet);
- Optimized towards NMT.
- `http://marian-nmt.github.io` and `https://github.com/marian-nmt/marian`

Part I

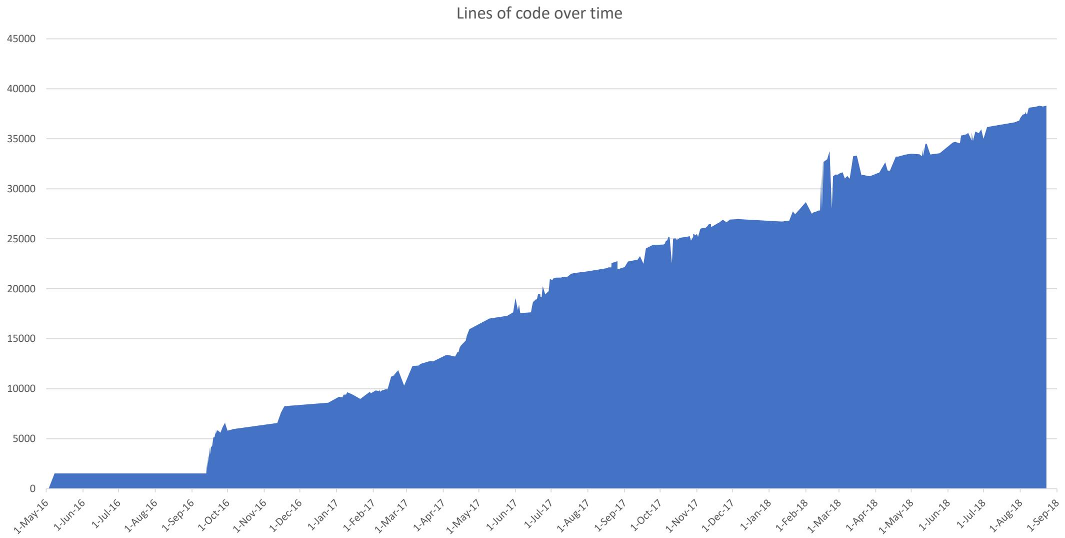
A Machine Translation Marathon 2016 project

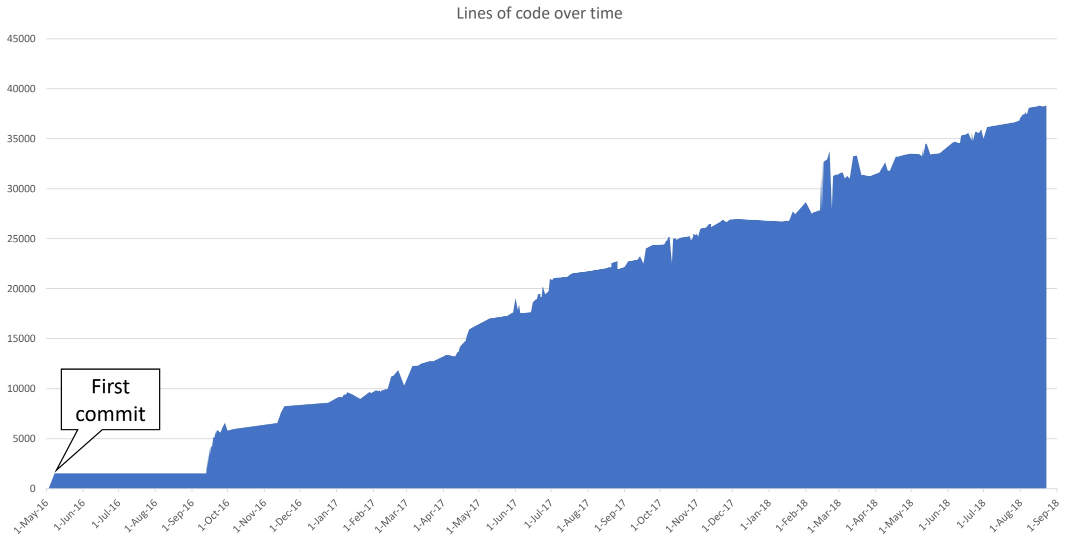
The first commit

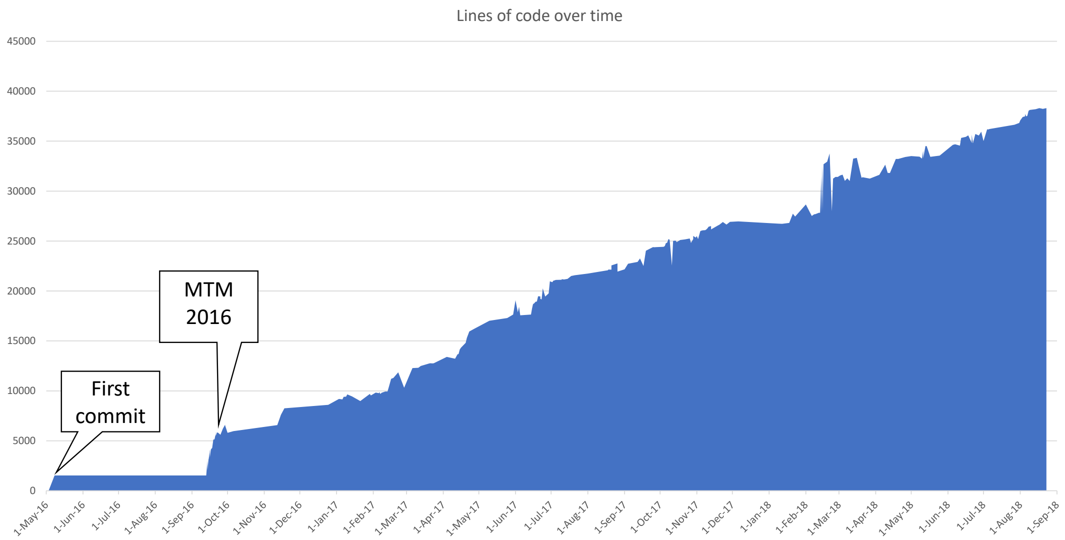
```
Commit:  6a7c93
Date:    May 4th, 2016
Message: very cool

Lines:   155
```

```cpp
#include <iostream>
#include "mad.h"

int main(int argc, char** argv) {
  Var x0 = 1, x1 = 2, x2 = 3;
  auto y = x0 + x0 + log(x2) + x1;

  std::vector<Var> x = { x0, x1, x2 };

  set_zero_all_adjoints();
  y.grad();

  std::cerr << "y = " << y.val() << std::endl;
  for(int i = 0; i < x.size(); ++i)
    std::cerr << "dy/dx_" << i << " = "
              << x[i].adj() << std::endl;
}
```

```
Var x0 = 1, x1 = 2, x2 = 3;
auto y = x0 + x0 + log(x2) + x1;

y = 5.09861
dy/dx_0 = 2
dy/dx_1 = 1
dy/dx_2 = 0.333333
```


Lines of code over time
45000
40000
35000
30000
25000
20000
15000
10000
5000
0
1-May-16
1-Jun-16
1-Jul-16
1-Aug-16
1-Sep-16
1-Oct-16
1-Nov-16
1-Dec-16
1-Jan-17
1-Feb-17
1-Mar-17
1-Apr-17
1-May-17
1-Jun-17
1-Jul-17
1-Aug-17
1-Sep-17
1-Oct-17
1-Nov-17
1-Dec-17
1-Jan-18
1-Feb-18
1-Mar-18
1-Apr-18
1-May-18
1-Jun-18
1-Jul-18
1-Aug-18
1-Sep-18


Lines of code over time
45000
40000
35000
30000
25000
20000
15000
10000
5000
0
1-May-16
1-Jun-16
1-Jul-16
1-Aug-16
1-Sep-16
1-Oct-16
1-Nov-16
1-Dec-16
1-Jan-17
1-Feb-17
1-Mar-17
1-Apr-17
1-May-17
1-Jun-17
1-Jul-17
1-Aug-17
1-Sep-17
1-Oct-17
1-Nov-17
1-Dec-17
1-Jan-18
1-Feb-18
1-Mar-18
1-Apr-18
1-May-18
1-Jun-18
1-Jul-18
1-Aug-18
1-Sep-18
First commit


Lines of code over time
45000
40000
35000
30000
25000
20000
15000
10000
5000
0
1-May-16
1-Jun-16
1-Jul-16
1-Aug-16
1-Sep-16
1-Oct-16
1-Nov-16
1-Dec-16
1-Jan-17
1-Feb-17
1-Mar-17
1-Apr-17
1-May-17
1-Jun-17
1-Jul-17
1-Aug-17
1-Sep-17
1-Oct-17
1-Nov-17
1-Dec-17
1-Jan-18
1-Feb-18
1-Mar-18
1-Apr-18
1-May-18
1-Jun-18
1-Jul-18
1-Aug-18
1-Sep-18
First commit
MTM 2016

A Neural Network Toolkit for MT

Maximiliana Behnke
Tomasz Dwojak
Marcin Junczys-Dowmunt
Roman Grundkiewicz
Andre Martins
Hieu Hoang
Lane Schwartz

A Neural Network Toolkit ~~for MT~~

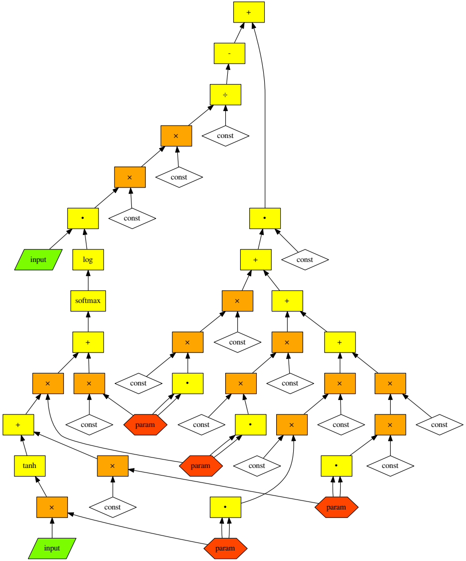
Maximiliana Behnke
Tomasz Dwojak
Marcin Junczys-Dowmunt
Roman Grundkiewicz
Andre Martins
Hieu Hoang
Lane Schwartz

Why create another NN toolkit?

- Flexibility
  - Add functionality easier & faster
- Speed
  - Pure C++ implementation
  - GPU-enabled (CPU may come soon)
- Learn about Deep Learning
  - Implement everything from scratch by ourselves

What we've achieved this week?

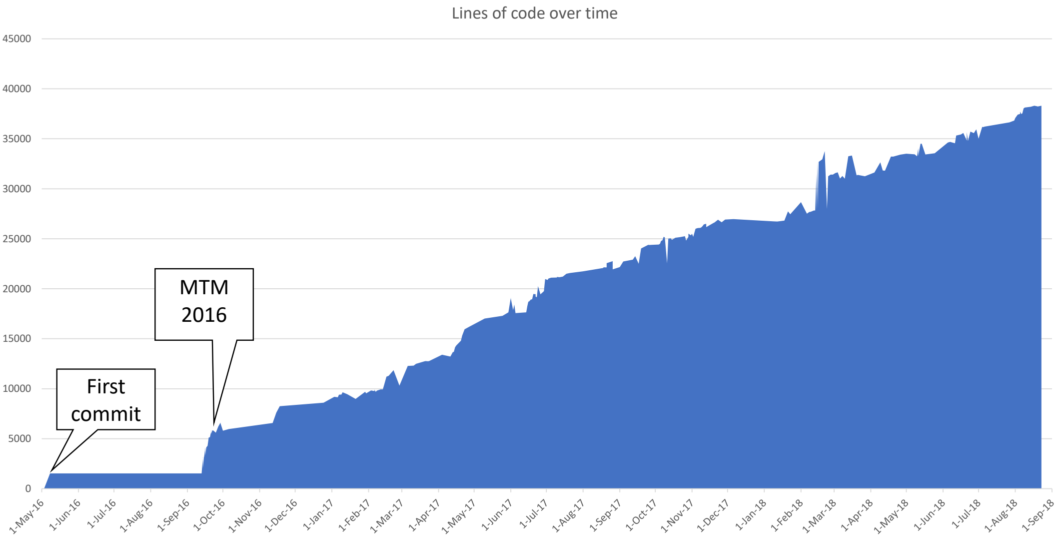
- Framework to create computation graphs
  - Simple feedforward NN
  - RNN, GRU, LSTM…
  - Binary or multiclass classifier
- Forward step
  - Classify, given input data and weights
- Backward step
  - Learn weights using backpropagation
- Tested with small datasets
  - MNIST (digit image recognition task)
  - MT
- Documentation
  - Doxygen

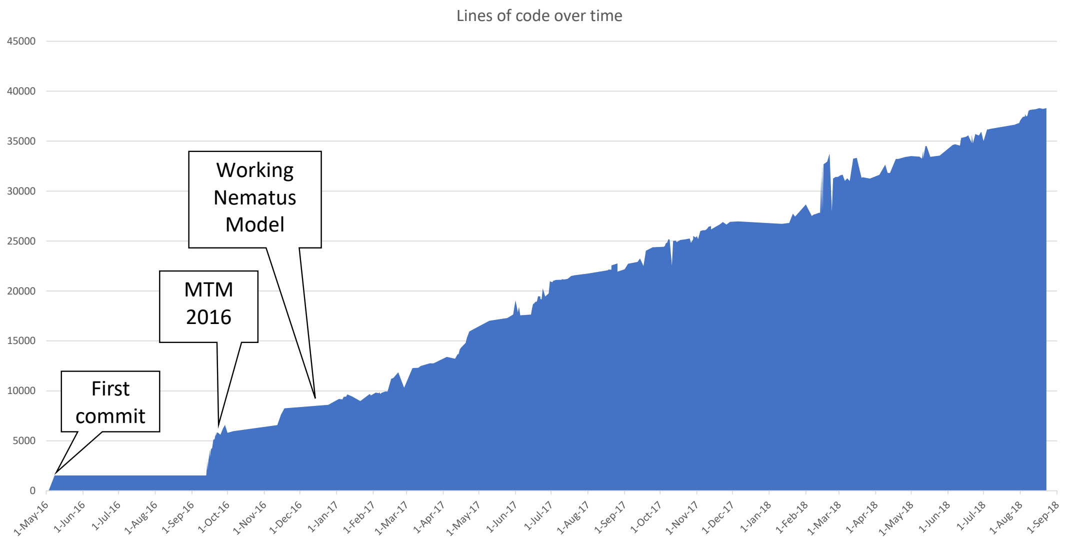


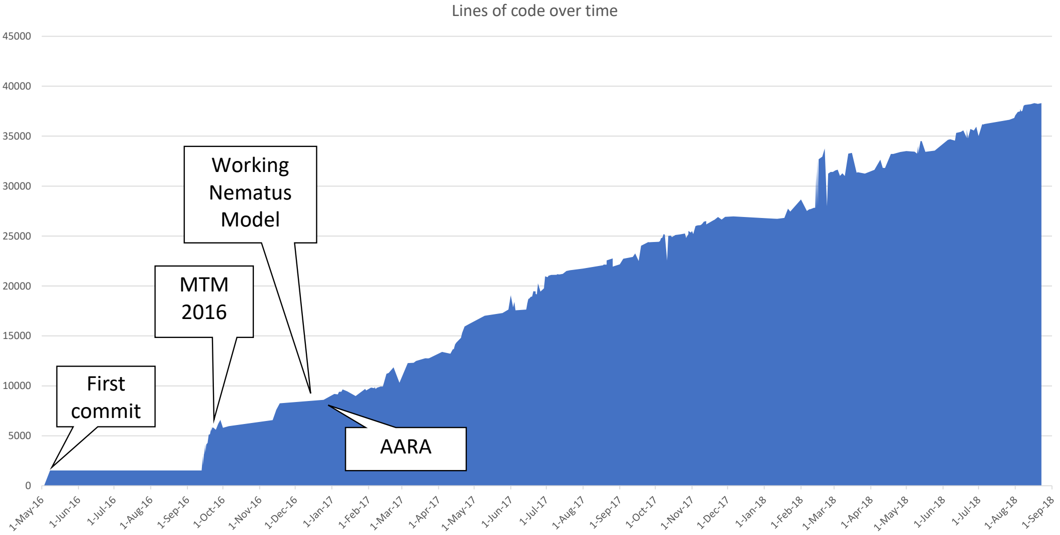
+
-
÷
×
const
•
input
log
softmax
tanh
param

What needs to be finished?

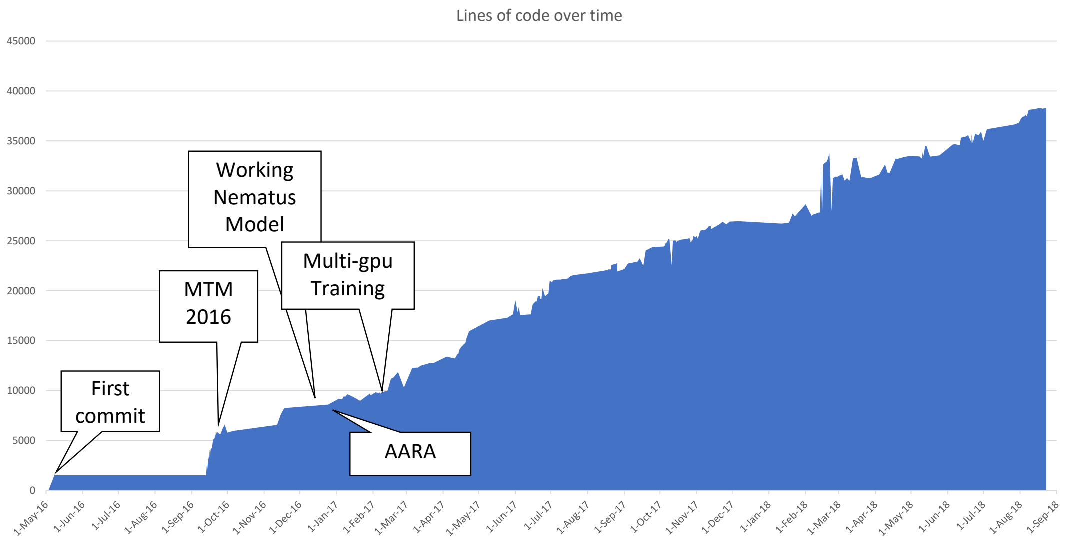
- Basic features:
  - Data shuffling
  - More random distributions
  - ...
- Model serialization & deserialization
- Documentation
- ...

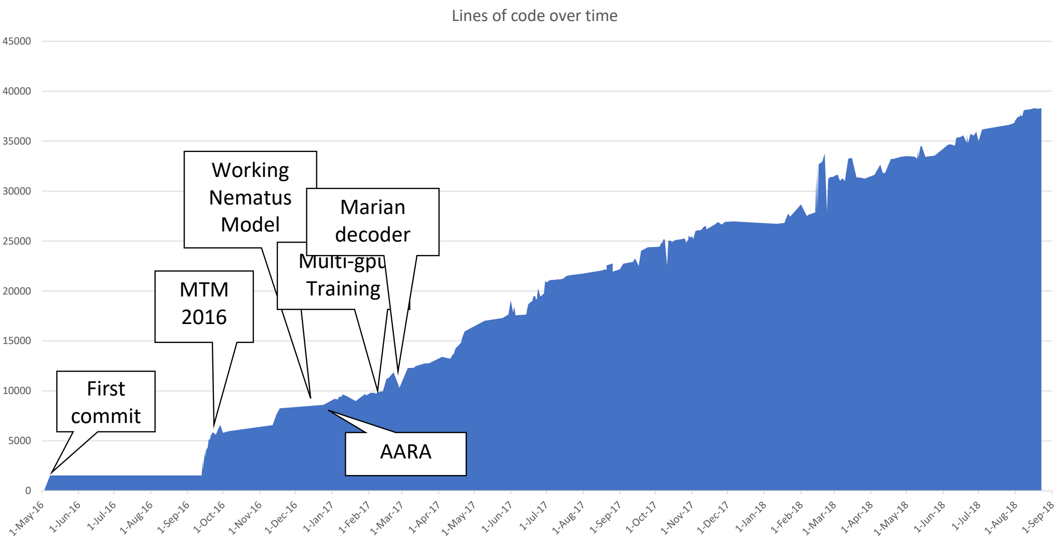

Lines of code over time
45000
40000
35000
30000
25000
20000
15000
10000
5000
0
1-May-16
1-Jun-16
1-Jul-16
1-Aug-16
1-Sep-16
1-Oct-16
1-Nov-16
1-Dec-16
1-Jan-17
1-Feb-17
1-Mar-17
1-Apr-17
1-May-17
1-Jun-17
1-Jul-17
1-Aug-17
1-Sep-17
1-Oct-17
1-Nov-17
1-Dec-17
1-Jan-18
1-Feb-18
1-Mar-18
1-Apr-18
1-May-18
1-Jun-18
1-Jul-18
1-Aug-18
1-Sep-18
First commit
MTM 2016


Lines of code over time
45000
40000
35000
30000
25000
20000
15000
10000
5000
0
1-May-16
1-Jun-16
1-Jul-16
1-Aug-16
1-Sep-16
1-Oct-16
1-Nov-16
1-Dec-16
1-Jan-17
1-Feb-17
1-Mar-17
1-Apr-17
1-May-17
1-Jun-17
1-Jul-17
1-Aug-17
1-Sep-17
1-Oct-17
1-Nov-17
1-Dec-17
1-Jan-18
1-Feb-18
1-Mar-18
1-Apr-18
1-May-18
1-Jun-18
1-Jul-18
1-Aug-18
1-Sep-18
First commit
MTM 2016
Working Nematus Model

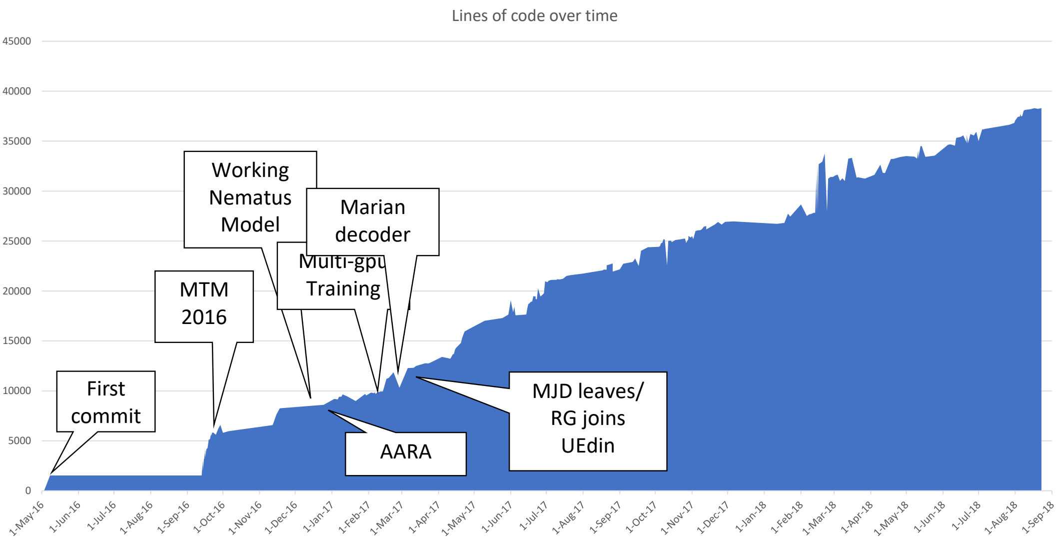


Lines of code over time
45000
40000
35000
30000
25000
20000
15000
10000
5000
0
1-May-16
1-Jun-16
1-Jul-16
1-Aug-16
1-Sep-16
1-Oct-16
1-Nov-16
1-Dec-16
1-Jan-17
1-Feb-17
1-Mar-17
1-Apr-17
1-May-17
1-Jun-17
1-Jul-17
1-Aug-17
1-Sep-17
1-Oct-17
1-Nov-17
1-Dec-17
1-Jan-18
1-Feb-18
1-Mar-18
1-Apr-18
1-May-18
1-Jun-18
1-Jul-18
1-Aug-18
1-Sep-18
First commit
MTM 2016
Working Nematus Model
AARA

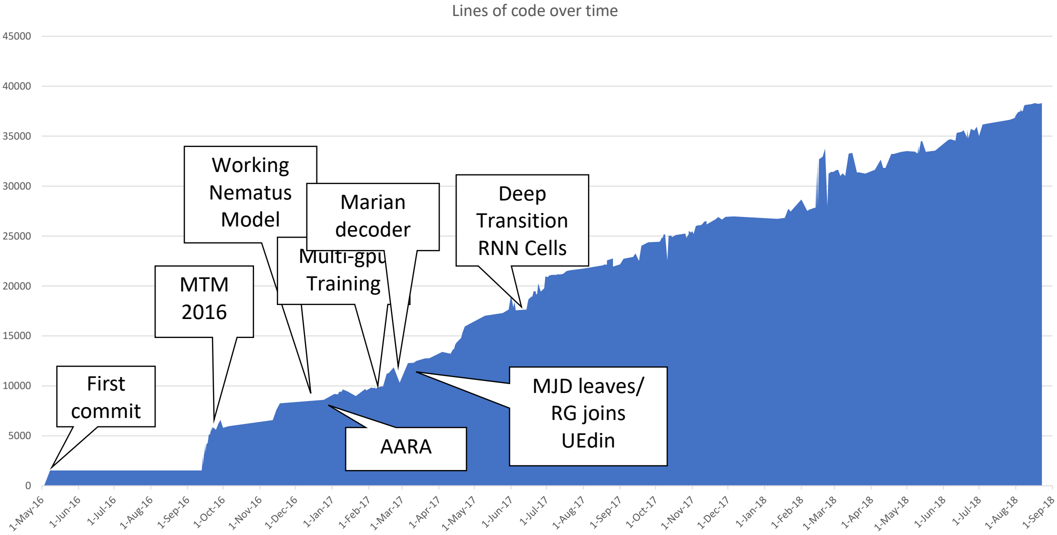

Lines of code over time
45000
40000
35000
30000
25000
20000
15000
10000
5000
0
1-May-16
1-Jun-16
1-Jul-16
1-Aug-16
1-Sep-16
1-Oct-16
1-Nov-16
1-Dec-16
1-Jan-17
1-Feb-17
1-Mar-17
1-Apr-17
1-May-17
1-Jun-17
1-Jul-17
1-Aug-17
1-Sep-17
1-Oct-17
1-Nov-17
1-Dec-17
1-Jan-18
1-Feb-18
1-Mar-18
1-Apr-18
1-May-18
1-Jun-18
1-Jul-18
1-Aug-18
1-Sep-18
First commit
MTM 2016
Working Nematus Model
Multi-gpu Training
AARA



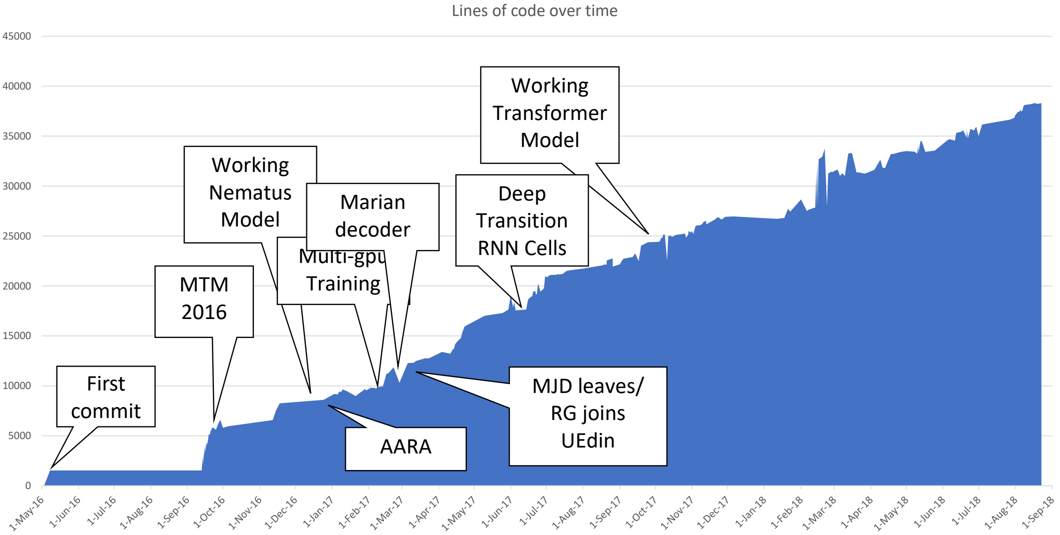
Lines of code over time
45000
40000
35000
30000
25000
20000
15000
10000
5000
0
1-May-16
1-Jun-16
1-Jul-16
1-Aug-16
1-Sep-16
1-Oct-16
1-Nov-16
1-Dec-16
1-Jan-17
1-Feb-17
1-Mar-17
1-Apr-17
1-May-17
1-Jun-17
1-Jul-17
1-Aug-17
1-Sep-17
1-Oct-17
1-Nov-17
1-Dec-17
1-Jan-18
1-Feb-18
1-Mar-18
1-Apr-18
1-May-18
1-Jun-18
1-Jul-18
1-Aug-18
1-Sep-18
First commit
MTM 2016
Working Nematus Model
Multi-gpu Training
Marian decoder
AARA

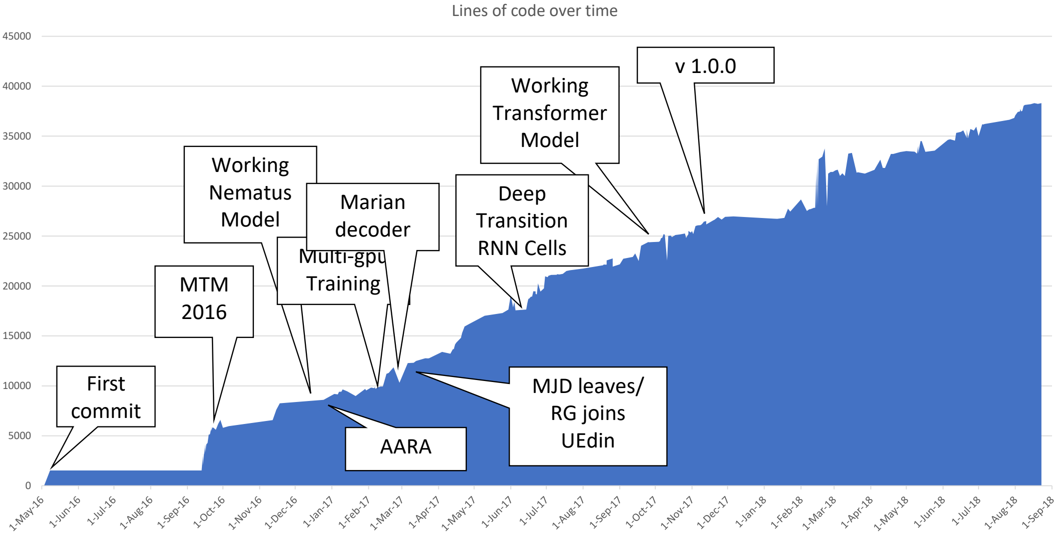


Lines of code over time
45000
40000
35000
30000
25000
20000
15000
10000
5000
0
1-May-16
1-Jun-16
1-Jul-16
1-Aug-16
1-Sep-16
1-Oct-16
1-Nov-16
1-Dec-16
1-Jan-17
1-Feb-17
1-Mar-17
1-Apr-17
1-May-17
1-Jun-17
1-Jul-17
1-Aug-17
1-Sep-17
1-Oct-17
1-Nov-17
1-Dec-17
1-Jan-18
1-Feb-18
1-Mar-18
1-Apr-18
1-May-18
1-Jun-18
1-Jul-18
1-Aug-18
1-Sep-18
First commit
MTM 2016
Working Nematus Model
Multi-gpu Training
Marian decoder
AARA
MJD leaves/ RG joins UEdin

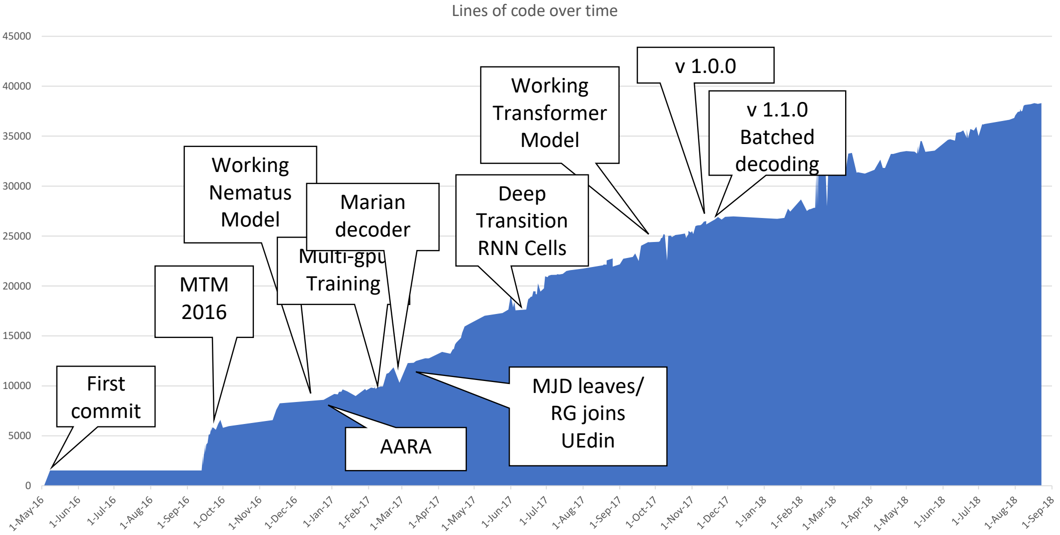


Lines of code over time
45000
40000
35000
30000
25000
20000
15000
10000
5000
0
1-May-16
1-Jun-16
1-Jul-16
1-Aug-16
1-Sep-16
1-Oct-16
1-Nov-16
1-Dec-16
1-Jan-17
1-Feb-17
1-Mar-17
1-Apr-17
1-May-17
1-Jun-17
1-Jul-17
1-Aug-17
1-Sep-17
1-Oct-17
1-Nov-17
1-Dec-17
1-Jan-18
1-Feb-18
1-Mar-18
1-Apr-18
1-May-18
1-Jun-18
1-Jul-18
1-Aug-18
1-Sep-18
First commit
MTM 2016
Working Nematus Model
Multi-gpu Training
Marian decoder
AARA
Deep Transition RNN Cells
MJD leaves/ RG joins UEdin

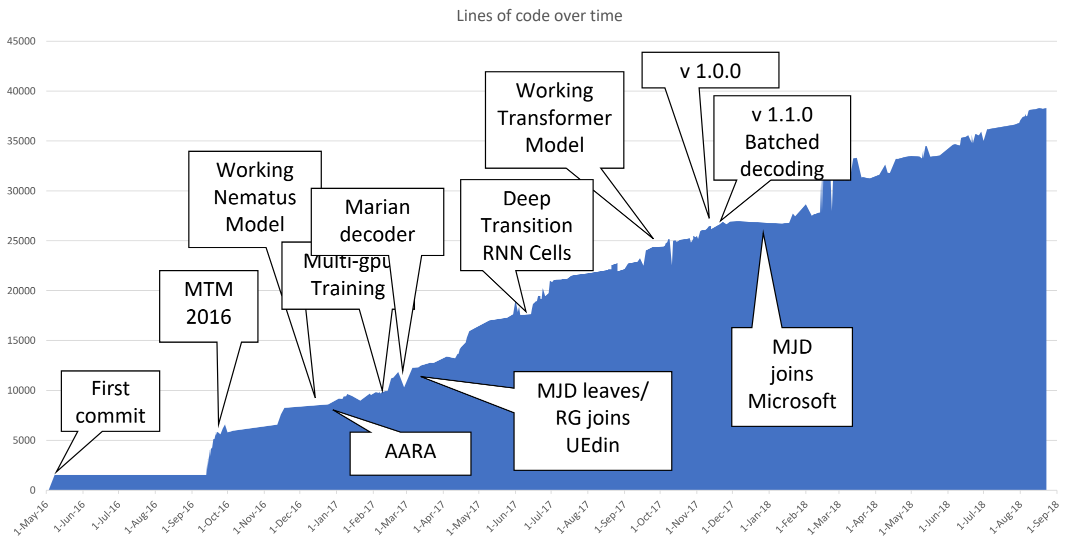

Lines of code over time
45000
40000
35000
30000
25000
20000
15000
10000
5000
0
1-May-16
1-Jun-16
1-Jul-16
1-Aug-16
1-Sep-16
1-Oct-16
1-Nov-16
1-Dec-16
1-Jan-17
1-Feb-17
1-Mar-17
1-Apr-17
1-May-17
1-Jun-17
1-Jul-17
1-Aug-17
1-Sep-17
1-Oct-17
1-Nov-17
1-Dec-17
1-Jan-18
1-Feb-18
1-Mar-18
1-Apr-18
1-May-18
1-Jun-18
1-Jul-18
1-Aug-18
1-Sep-18
First commit
MTM 2016
Working Nematus Model
Multi-gpu Training
Marian decoder
AARA
Deep Transition RNN Cells
Working Transformer Model
MJD leaves/ RG joins UEdin

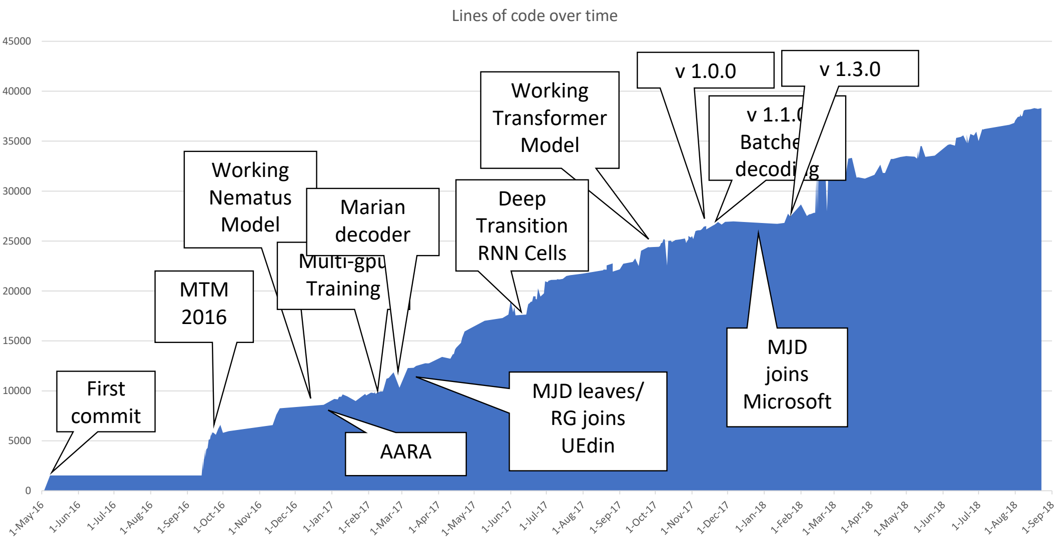


Lines of code over time
45000
40000
35000
30000
25000
20000
15000
10000
5000
0
1-May-16
1-Jun-16
1-Jul-16
1-Aug-16
1-Sep-16
1-Oct-16
1-Nov-16
1-Dec-16
1-Jan-17
1-Feb-17
1-Mar-17
1-Apr-17
1-May-17
1-Jun-17
1-Jul-17
1-Aug-17
1-Sep-17
1-Oct-17
1-Nov-17
1-Dec-17
1-Jan-18
1-Feb-18
1-Mar-18
1-Apr-18
1-May-18
1-Jun-18
1-Jul-18
1-Aug-18
1-Sep-18
First commit
MTM 2016
Working Nematus Model
Multi-gpu Training
Marian decoder
AARA
MJD leaves/ RG joins UEdin
Deep Transition RNN Cells
Working Transformer Model
v 1.0.0

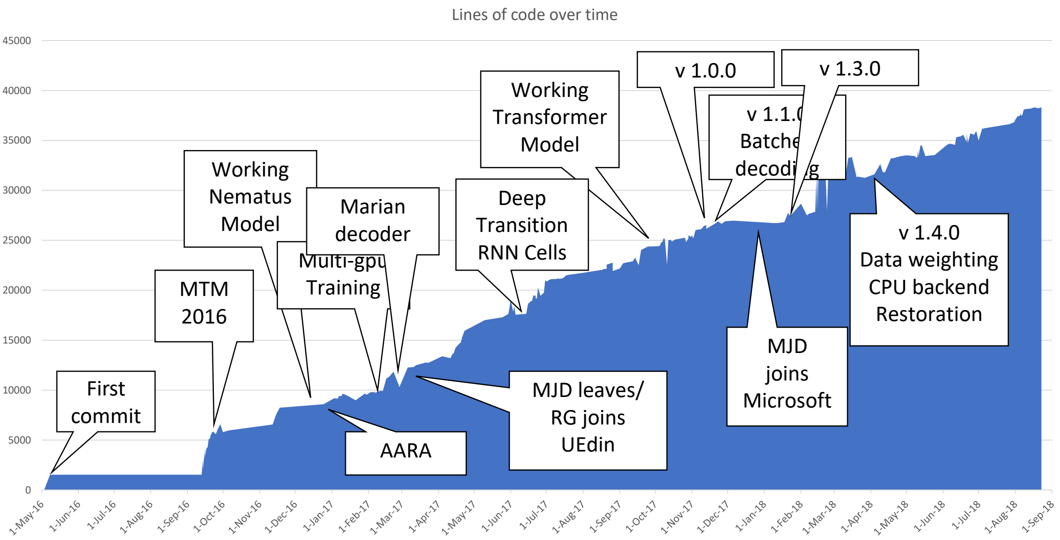


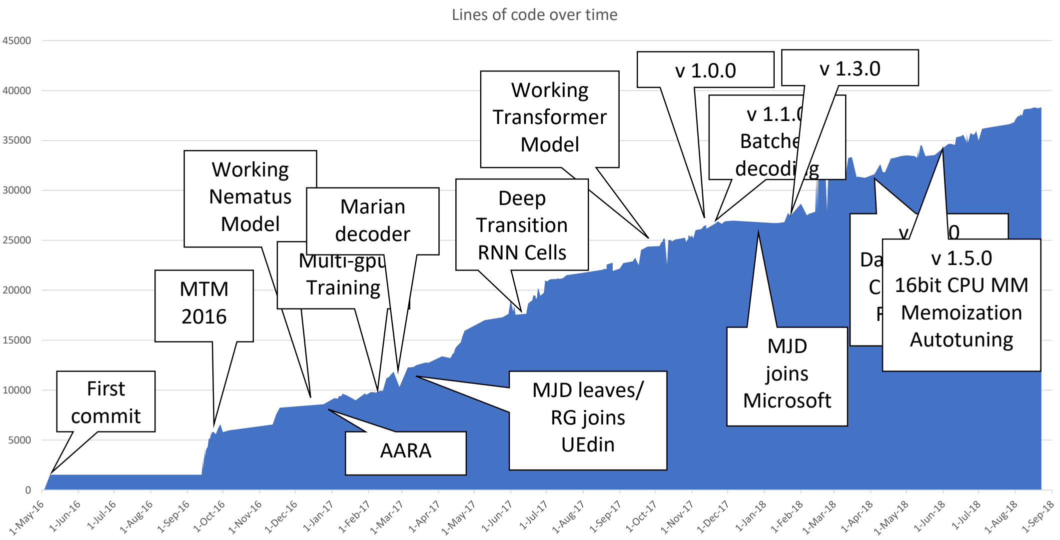
Lines of code over time
45000
40000
35000
30000
25000
20000
15000
10000
5000
0
1-May-16
1-Jun-16
1-Jul-16
1-Aug-16
1-Sep-16
1-Oct-16
1-Nov-16
1-Dec-16
1-Jan-17
1-Feb-17
1-Mar-17
1-Apr-17
1-May-17
1-Jun-17
1-Jul-17
1-Aug-17
1-Sep-17
1-Oct-17
1-Nov-17
1-Dec-17
1-Jan-18
1-Feb-18
1-Mar-18
1-Apr-18
1-May-18
1-Jun-18
1-Jul-18
1-Aug-18
1-Sep-18
First commit
MTM 2016
Working Nematus Model
Marian decoder
Multi-gpu Training
AARA
MJD leaves/ RG joins UEdin
Deep Transition RNN Cells
Working Transformer Model
v 1.0.0
v 1.1.0 Batched decoding

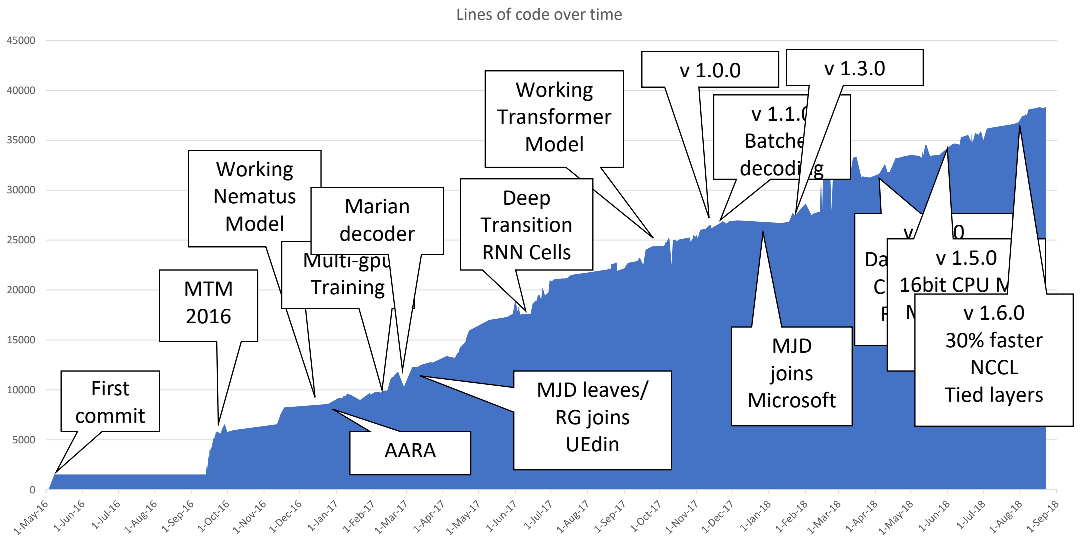

Lines of code over time
45000
40000
35000
30000
25000
20000
15000
10000
5000
0
1-May-16
1-Jun-16
1-Jul-16
1-Aug-16
1-Sep-16
1-Oct-16
1-Nov-16
1-Dec-16
1-Jan-17
1-Feb-17
1-Mar-17
1-Apr-17
1-May-17
1-Jun-17
1-Jul-17
1-Aug-17
1-Sep-17
1-Oct-17
1-Nov-17
1-Dec-17
1-Jan-18
1-Feb-18
1-Mar-18
1-Apr-18
1-May-18
1-Jun-18
1-Jul-18
1-Aug-18
1-Sep-18
First commit
MTM 2016
Working Nematus Model
Multi-gpu Training
Marian decoder
AARA
MJD leaves/ RG joins UEdin
Deep Transition RNN Cells
Working Transformer Model
v 1.0.0
v 1.1.0 Batched decoding
MJD joins Microsoft



Lines of code over time
45000
40000
35000
30000
25000
20000
15000
10000
5000
0
1-May-16
1-Jun-16
1-Jul-16
1-Aug-16
1-Sep-16
1-Oct-16
1-Nov-16
1-Dec-16
1-Jan-17
1-Feb-17
1-Mar-17
1-Apr-17
1-May-17
1-Jun-17
1-Jul-17
1-Aug-17
1-Sep-17
1-Oct-17
1-Nov-17
1-Dec-17
1-Jan-18
1-Feb-18
1-Mar-18
1-Apr-18
1-May-18
1-Jun-18
1-Jul-18
1-Aug-18
1-Sep-18
First commit
MTM 2016
Working Nematus Model
Multi-gpu Training
Marian decoder
AARA
MJD leaves/ RG joins UEdin
Deep Transition RNN Cells
Working Transformer Model
v 1.0.0
v 1.3.0
MJD joins Microsoft


Lines of code over time
45000
40000
35000
30000
25000
20000
15000
10000
5000
0
1-May-16
1-Jun-16
1-Jul-16
1-Aug-16
1-Sep-16
1-Oct-16
1-Nov-16
1-Dec-16
1-Jan-17
1-Feb-17
1-Mar-17
1-Apr-17
1-May-17
1-Jun-17
1-Jul-17
1-Aug-17
1-Sep-17
1-Oct-17
1-Nov-17
1-Dec-17
1-Jan-18
1-Feb-18
1-Mar-18
1-Apr-18
1-May-18
1-Jun-18
1-Jul-18
1-Aug-18
1-Sep-18
First commit
MTM 2016
Working Nematus Model
Multi-gpu Training
Marian decoder
AARA
MJD leaves/ RG joins UEdin
Deep Transition RNN Cells
Working Transformer Model
v 1.0.0
v 1.1.0 Batched decoding
v 1.3.0
MJD joins Microsoft
v 1.4.0 Data weighting CPU backend Restoration

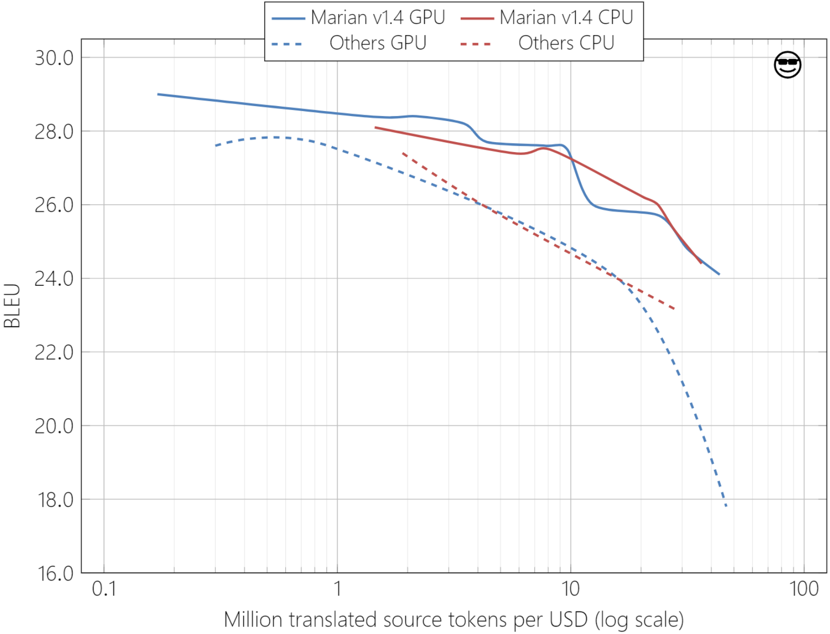

Lines of code over time
45000
40000
35000
30000
25000
20000
15000
10000
5000
0
1-May-16
1-Jun-16
1-Jul-16
1-Aug-16
1-Sep-16
1-Oct-16
1-Nov-16
1-Dec-16
1-Jan-17
1-Feb-17
1-Mar-17
1-Apr-17
1-May-17
1-Jun-17
1-Jul-17
1-Aug-17
1-Sep-17
1-Oct-17
1-Nov-17
1-Dec-17
1-Jan-18
1-Feb-18
1-Mar-18
1-Apr-18
1-May-18
1-Jun-18
1-Jul-18
1-Aug-18
1-Sep-18
First commit
MTM 2016
Working Nematus Model
Multi-gpu Training
Marian decoder
AARA
Deep Transition RNN Cells
MJD leaves/ RG joins UEdin
Working Transformer Model
v 1.0.0
v 1.3.0
MJD joins Microsoft
v 1.5.0
16bit CPU MM
Memoization
Autotuning



Lines of code over time
45000
40000
35000
30000
25000
20000
15000
10000
5000
0
1-May-16
1-Jun-16
1-Jul-16
1-Aug-16
1-Sep-16
1-Oct-16
1-Nov-16
1-Dec-16
1-Jan-17
1-Feb-17
1-Mar-17
1-Apr-17
1-May-17
1-Jun-17
1-Jul-17
1-Aug-17
1-Sep-17
1-Oct-17
1-Nov-17
1-Dec-17
1-Jan-18
1-Feb-18
1-Mar-18
1-Apr-18
1-May-18
1-Jun-18
1-Jul-18
1-Aug-18
1-Sep-18
First commit
MTM 2016
Working Nematus Model
Marian decoder
Multi-gpu Training
AARA
Deep Transition RNN Cells
Working Transformer Model
MJD leaves/ RG joins UEdin
v 1.0.0
v 1.3.0
MJD joins Microsoft
v 1.5.0
v 1.6.0 30% faster NCCL Tied layers

Going further

- Reduce dependencies for CPU version to zero
- Reduce dependencies for GPU version to CUDA
- Become faster and more versatile
- Research tool with immediate deployment


Marian v1.4 GPU
Marian v1.4 CPU
Others GPU
Others CPU
30.0
28.0
26.0
24.0
22.0
20.0
18.0
16.0
BLEU
0.1
1
10
100
Million translated source tokens per USD (log scale)

System Output List

English-German newstest2018

Interested in Contributing?

- Check out available resources.
- Create an account and start submitting your own systems.

Scored Systems

System	Submitter	System Notes	Constraint	Run Notes	BLEU	BLEU-cased	TER	BEER 2.0	CharactTER
Marian-Transformer (*Details*)	marcinjd *Microsoft*	Marian Transformer-Big	yes	Transformer-big ensemble x4. With back-translation, data-filtering on Paracrawl with domain-weighting. Decoder-time ensembling with transformer-LM, right-to-left decoding.	48.9	**48.3**	0.407	0.697	0.362
NMT-SMT Hybrid (*Details*)	fstahlberg *University of Cambridge*	MBR-based combination of neural models and SMT	yes		47.1	46.6	0.415	0.691	0.369
NTT Transformer-based System (*Details*)	makoto-mr *NTT*	Based on Transformer Big model. Trained with filtered version of CommonCrawl, ParaCrawl and synthetic corpus of newscrawl2017. R2L reranking.	yes		47.0	46.5	0.426	0.688	0.370
Primary Submission (*Details*)	pianist *Karlsruhe Institute of Technology*	Primary Submission	yes		46.9	46.3	0.428	0.687	0.382
MMT-unconstraint (*Details*)	nicolabertoldi *MMT srl*		no		46.7	46.2	0.432	0.682	0.387
JHU (*Details*)	jhu-nmt *Johns Hopkins University*	Marian Deep RNN	yes	Marian deep model, ensemble of 4 runs using base data (without Paracrawl), re-back-translated news 2016. Not final system yet.	43.6	43.0	0.453	0.670	0.394
MMT-constraint-dec (*Details*)	nicolabertoldi *MMT srl*		yes		42.9	42.5	0.463	0.667	0.411
NJUNMT (*Details*)	ZhaoChengqi *Nanjing University*	transformer base without back translation	yes	transformer base without back translation	40.6	40.0	0.496	0.647	0.436
LMU-unsupervised-nmt-wmt18-en-de (*Details*)	Matthias.Huck *LMU Munich*	Unsupervised NMT (no parallel training corpora)	yes		15.8	15.5	0.762	0.500	*failed*
RWTH Unsupervised NMT Ensemble (*Details*)	yunsukim *RWTH Aachen University*	(Unsupervised) Transformer with shared encoder/decoder, separate top-50k word	yes		15.9	14.8	0.753	0.514	0.607

System Output List | screen short windows wir

matrix.statmt.org/matrix/systems_list/1881

System Output List

English-German newstest2018

Translations | Resources | Download | Info | Account

Interested in Contributing?

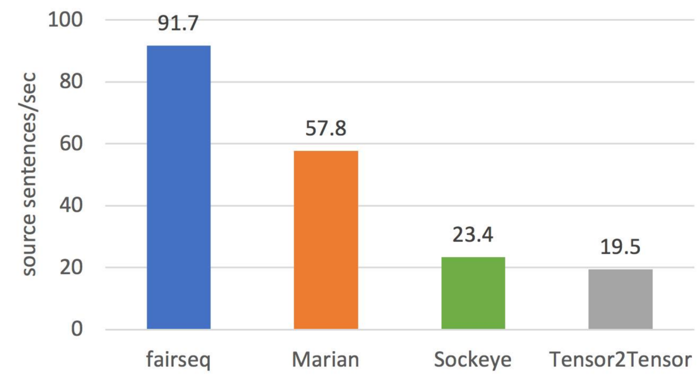
- Check out available resources.
- Create an account and start submitting your own systems.

Scored Systems

System	Submitter	System Notes	Constraint	Run Notes	BLEU	BLEU-cased	TER	BEER 2.0	CharactTER
Marian-Transformer *(Details)*	marcinjd *Microsoft*	Marian Transformer-Big	yes	Transformer-big ensemble x4. With back-translation, data-filtering on Paracrawl with domain-weighting. Decoder-time ensembling with transformer-LM, right-to-left decoding.	48.9	**48.3**	0.407	0.697	0.362
NMT-SMT Hybrid *(Details)*	fstahlberg *University of Cambridge*	MBR-based combination of neural models and SMT	yes		47.1	46.6	0.415	0.691	0.369
NTT Transformer-based System *(Details)*	makoto-mr *NTT*	Based on Transformer Big model. Trained with filtered version of CommonCrawl, ParaCrawl and synthetic corpus of newscrawl2017. R2L reranking.	yes		47.0	46.5	0.426	0.688	0.370
Primary Submission *(Details)*	pianist *Karlsruhe Institute of Technology*	Primary Submission	yes		46.9	46.3	0.428	0.687	0.382
MMT-unconstraint *(Details)*	nicolabertoldi *MMT srl*		no		46.7	46.2	0.432	0.682	0.387
JHU *(Details)*	jhu-nmt *Johns Hopkins University*	Marian Deep RNN	yes	Marian deep model, ensemble of 4 runs using base data (without Paracrawl), re-back-translated news 2016. Not final system yet.	43.6	43.0	0.453	0.670	0.394
MMT-constraint-dec *(Details)*	nicolabertoldi *MMT srl*		yes		42.9	42.5	0.463	0.667	0.411
NJUNMT *(Details)*	ZhaoChengqi *Nanjing University*	transformer base without back translation	yes	transformer base without back translation	40.6	40.0	0.496	0.647	0.436
LMU-unsupervised-nmt-wmt18-en-de *(Details)*	Matthias.Huck *LMU Munich*	Unsupervised NMT (no parallel training corpora)	yes		15.8	15.5	0.762	0.500	*failed*
RWTH Unsupervised NMT Ensemble *(Details)*	yunsukim *RWTH Aachen University*	(Unsupervised) Transformer with shared encoder/decoder, separate top-50k word	yes		15.9	14.8	0.753	0.514	0.607

Yann LeCun shared a post.
June 15

A major update of the FairSeq open source code:
- Transformer and Gated ConvNet models for translation.
- pre-trained models.
- parallel training and reduced precision.



Michael Auli
June 15 ·

We are releasing new features for fairseq, FAIR's sequence to sequence learning library: https://github.com/pytorch/fairseq

Distributed training, fp16, delayed batching
We release code and pre-trained models to reproduce our recent paper "Scaling Neural Machine Translation" (https://arxiv.org/abs/1806.00187) where we train on up to 128 GPUs with half precision floating point operations as well ... See More

133 · 49 Shares

Like · Comment · Share

Write a comment...

Fast inference
fairseq can generate translations at a rate of 92 sentences/sec for big Transformers on a fast GPU by clever caching, removing finished sentences from the computation and by batching by tokens. This improves speed by nearly 60%. The image shows a comparison to other toolkits (measured on a V100 GPU for WMT English-German translation on newstest2014 using a big Transformer).

Language models
fairseq now supports the training of gated convolutional language models (https://arxiv.org/abs/1612.08083). It can train a Google Billion Word language model on 128 GPUs in less than a day.

James Bradbury
@jekbradbury

Following

Facebook's fairseq MT engine is really, really fast... Like, 50% faster than @marian_nmt (which is itself way faster than Sockeye/OpenNMT/Tensor2Tensor/xnmt/Nematus/etc) at generating from the same Transformer model facebook.com/61013326/posts ...

2:24 PM - 15 Jun 2018

37 Retweets **139** Likes

2 37 139

Tweet your reply

Marian NMT @marian_nmt · Jun 15

Replying to @jekbradbury

Hold my beer ;)

1 13

James Bradbury
@jekbradbury

Following

Facebook's fairseq MT engine is really, really fast... Like, 50% faster than @marian_nmt (which is itself way faster than Sockeye/OpenNMT/Tensor2Tensor/xnmt/Nematus/etc) at generating from the same Transformer model facebook.com/61013326/posts ...

2:24 PM - 15 Jun 2018

37 Retweets **139** Likes

2 37 139

Tweet your reply

Marian NMT @marian_nmt · Jun 15

Replying to @jekbradbury

Hold my beer ;)

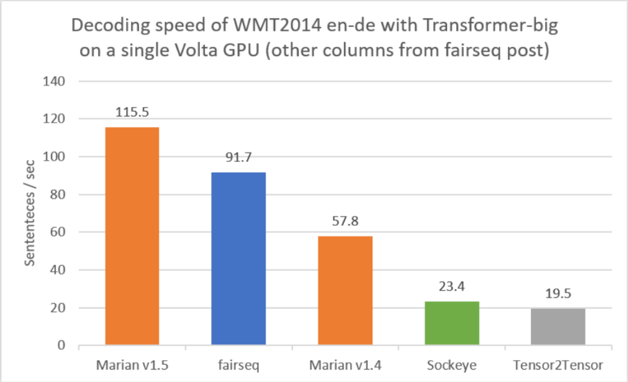
1 13

Marian NMT @marian_nmt · Jun 17

Boom! Marian v1.5.0 released.

Includes:
- Extensions from the WNMT shared task on efficiency arxiv.org/abs/1805.12096
- Optimized GPU-decoding for Transformer models.

See chart below for speed comparison to v1.4.0 (based on FAIR's post) @jekbradbury @alvations



4 33 108

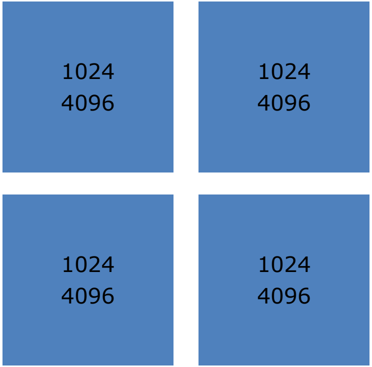
Part II

Decoding on the CPU

Quality first – speed later

- Lessons from WNMT shared task on efficient decoding;
- Sequence-level knowledge distillation (Kim & Rush 2016):
- Training four Transformer-big models on official task data (teacher);
- Translate entire EN data to DE-trans (8-best list);
- Select sentences with highest sentence-level BLEU based on DE-orig data;
- Train students on EN/DE-trans data.

Teacher
Transformer Big

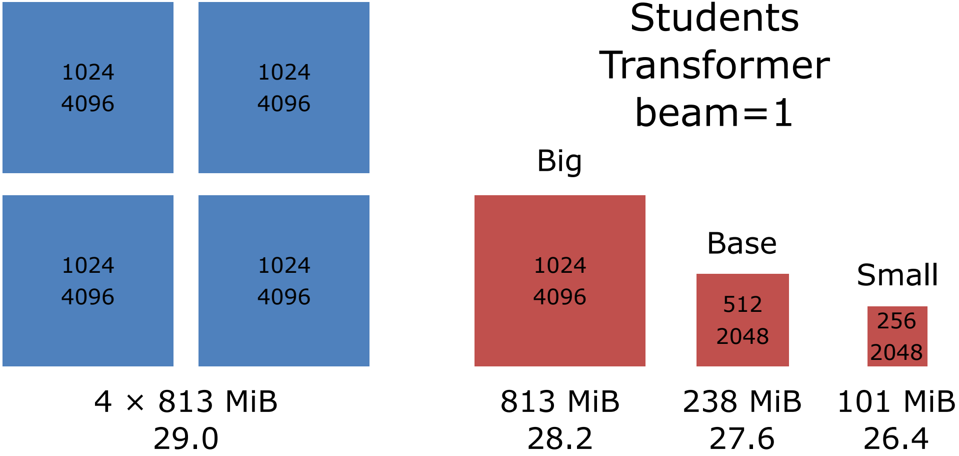


4 × 813 MiB
29.0

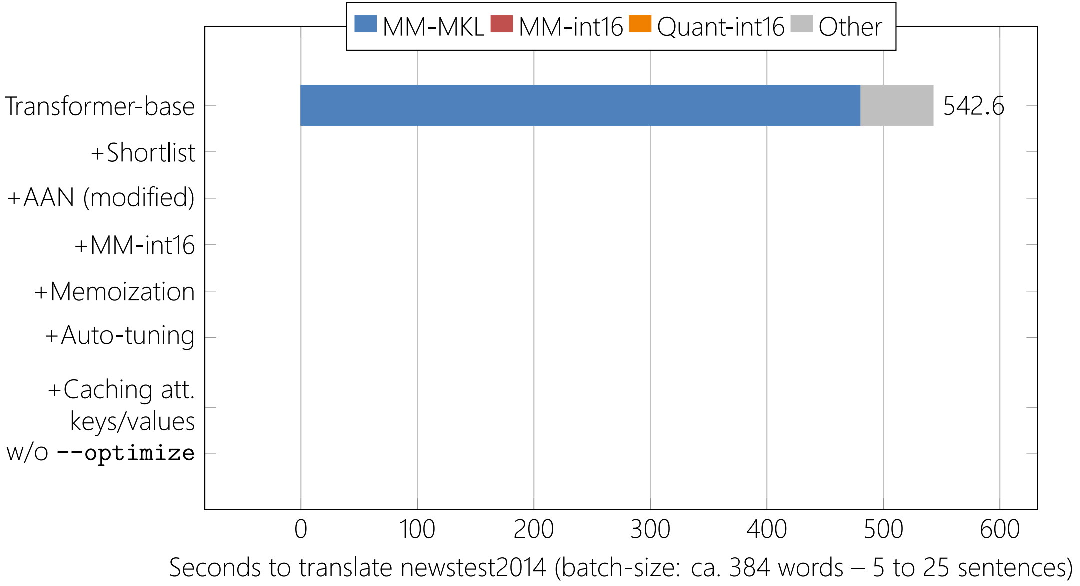
(Scale preserving)

Teacher
Transformer Big

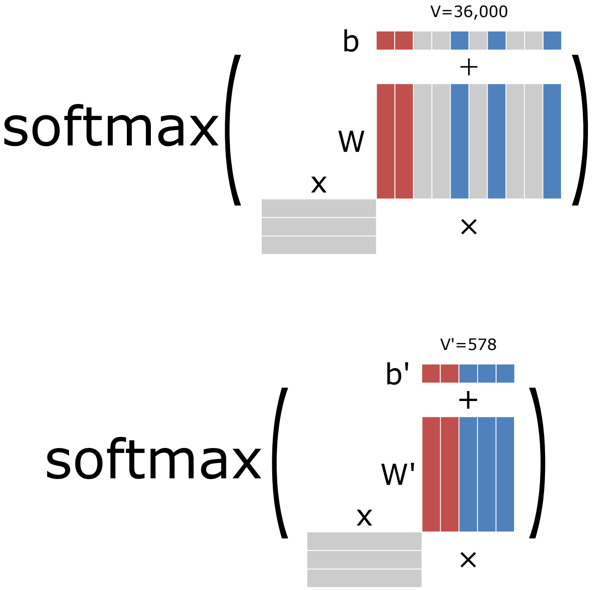
Students
Transformer
beam=1

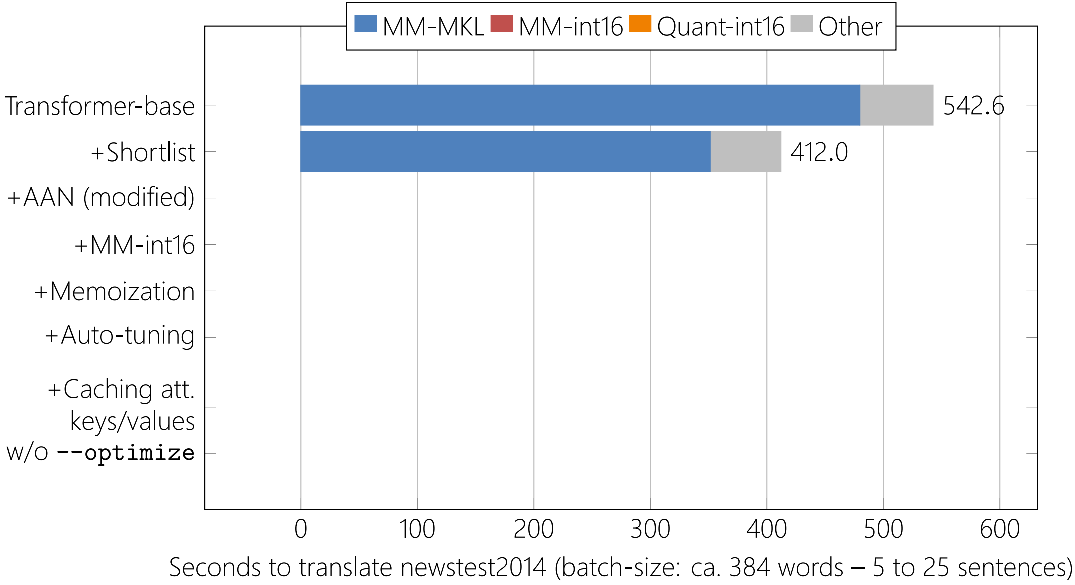


(Scale preserving)


MM-MKL
MM-int16
Quant-int16
Other
Transformer-base
+Shortlist
+AAN (modified)
+MM-int16
+Memoization
+Auto-tuning
+Caching att. keys/values
w/o --optimize
542.6
0
100
200
300
400
500
600
Seconds to translate newstest2014 (batch-size: ca. 384 words – 5 to 25 sentences)

</s>	dog Hund
<unk>	dog Hun@@
,	dog de@@
.	dog Hunde
the	...
in	lift heben
die	lift Li@@
of	lift ft
der	lift Aufzug
and	...




MM-MKL
MM-int16
Quant-int16
Other
Transformer-base
+Shortlist
+AAN (modified)
+MM-int16
+Memoization
+Auto-tuning
+Caching att. keys/values
w/o --optimize
542.6
412.0
0
100
200
300
400
500
600
Seconds to translate newstest2014 (batch-size: ca. 384 words – 5 to 25 sentences)

Multiplicative attention:

$$
\begin{aligned}
Q' &= QW_q + b_q \\
K' &= KW_k + b_k \\
V' &= VW_v + b_v \\
C &= \mathrm{softmax}(Q' \times (K')^T) \times V' \\
Y &= \mathrm{norm}(Q + C)
\end{aligned}
$$

- During training: $Q = K = V$
- During translation: $Q \neq K; K = V$
- Complexity per step: $O(n)$
- Because:
  $c_t = \mathrm{softmax}(q'_t \times (K'_{<t})^T) \times V'_{<t}$
  $K_{<t+1} = [K_{<t}; q_t]$

Multiplicative attention:

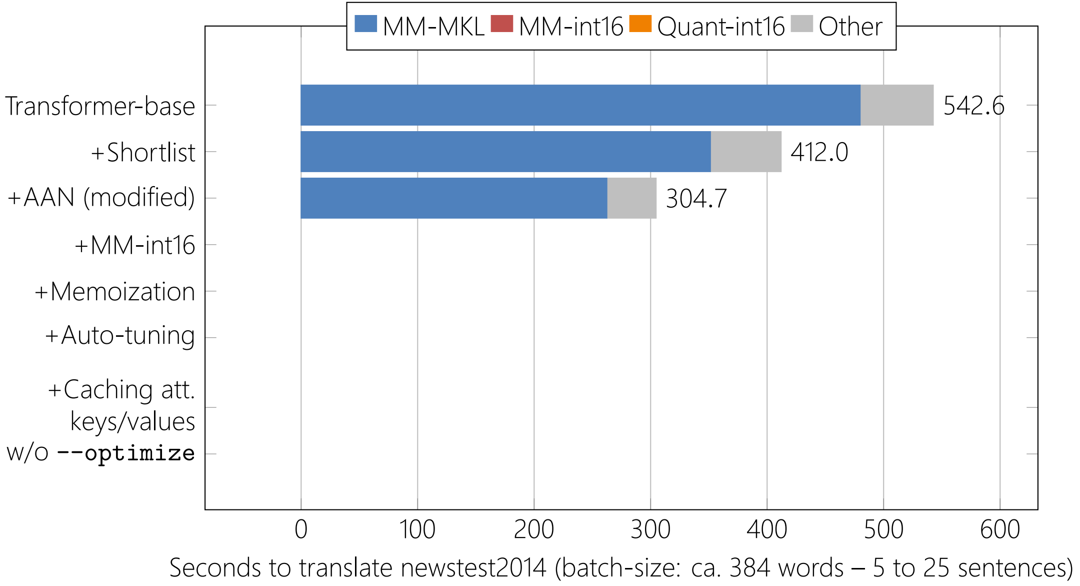
$$
\begin{aligned}
Q' &= QW_q + b_q \\
K' &= KW_k + b_k \\
V' &= VW_v + b_v \\
C &= \mathrm{softmax}(Q' \times (K')^T) \times V' \\
Y &= \mathrm{norm}(Q + C)
\end{aligned}
$$

- During training: $Q = K = V$
- During translation: $Q \neq K; K = V$
- Complexity per step: $O(n)$
- Because:
  $c_t = \mathrm{softmax}(q'_t \times (K'_{<t})^T) \times V'_{<t}$
  $K_{<t+1} = [K_{<t}; q_t]$

Average attention network (Zhang et al. 2018):

$$
\begin{aligned}
C &= \mathrm{gate}(\mathrm{FFN}(\bar{V}), Q) \\
Y &= \mathrm{norm}(Q + C)
\end{aligned}
$$

- Gate and FFN optional
- Complexity per step: $O(1)$
- Because: $\bar{v}_t = \frac{1}{t}((t-1)\bar{v}_{t-1} + v_t)$
- Basically a weird RNN
- Authors report 4x speed-up for beam=4



MM-MKL
MM-int16
Quant-int16
Other
Transformer-base
+Shortlist
+AAN (modified)
+MM-int16
+Memoization
+Auto-tuning
+Caching att. keys/values
w/o --optimize
542.6
412.0
304.7
0
100
200
300
400
500
600
Seconds to translate newstest2014 (batch-size: ca. 384 words – 5 to 25 sentences)

Code based on Devlin (2017), extended to AVX512

$$q(x) = \text{int16}(x \cdot 2^{10})$$

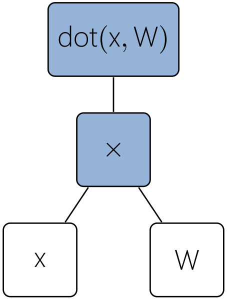
Code based on Devlin (2017), extended to AVX512

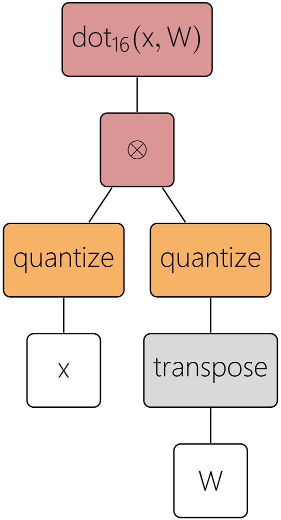
$$\begin{aligned} q(x) &= \mathrm{int16}(x \cdot 2^{10}) \\ A_q \otimes B_q &= A_q \times B_q^T \end{aligned}$$

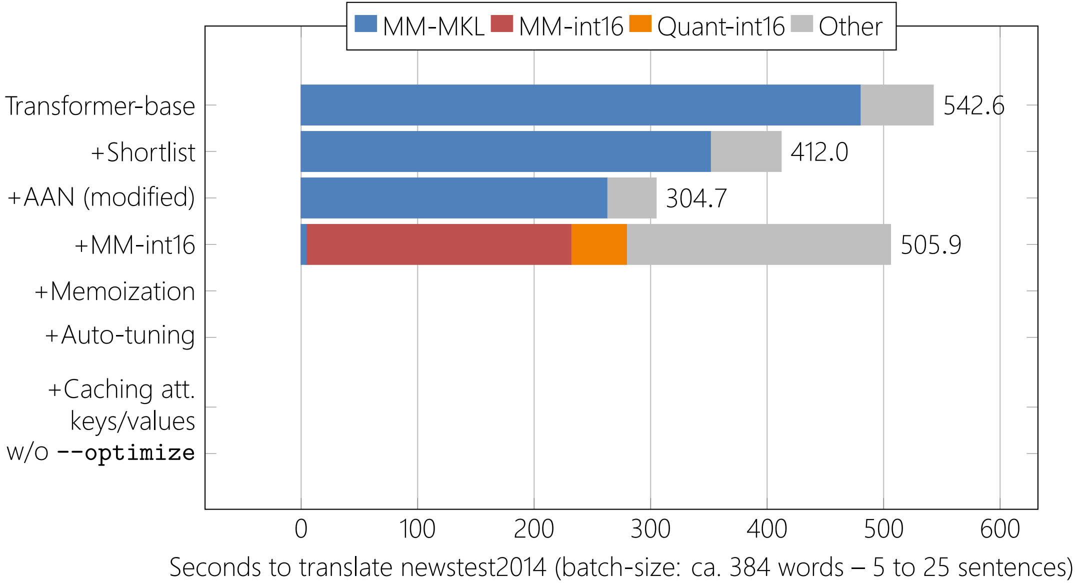
Code based on Devlin (2017), extended to AVX512

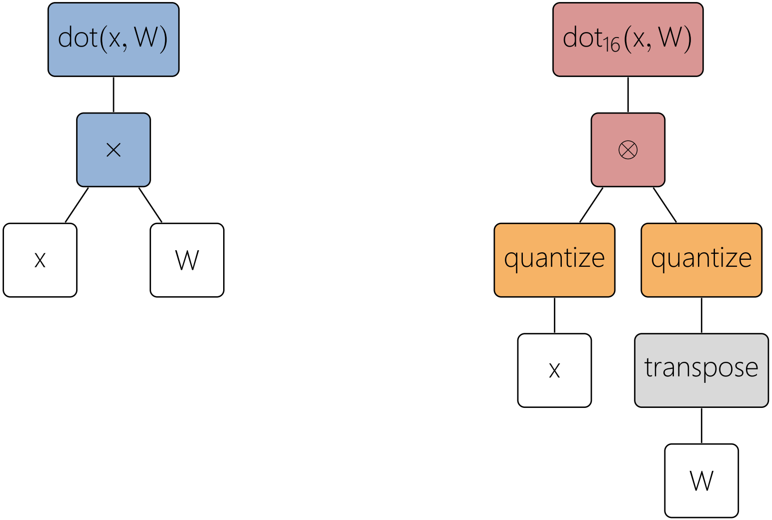
$$
\begin{aligned}
q(x) &= \mathrm{int16}(x \cdot 2^{10}) \\
A_q \otimes B_q &= A_q \times B_q^T
\end{aligned}
$$

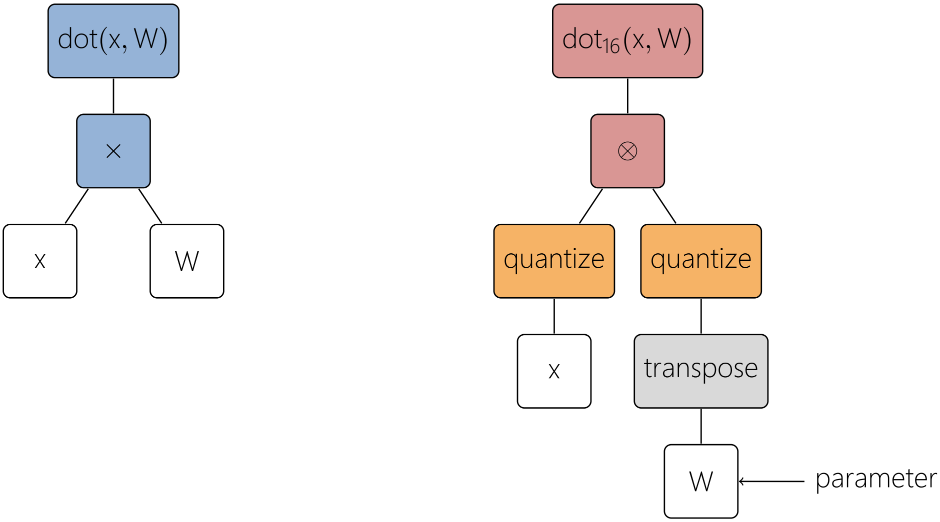
$$
\begin{aligned}
x \times W &= x \times (W^T)^T \\
&\approx q(x) \times q(W^T)^T \\
&= q(x) \otimes q(W^T)
\end{aligned}
$$


dot(x, W)
×
x
W

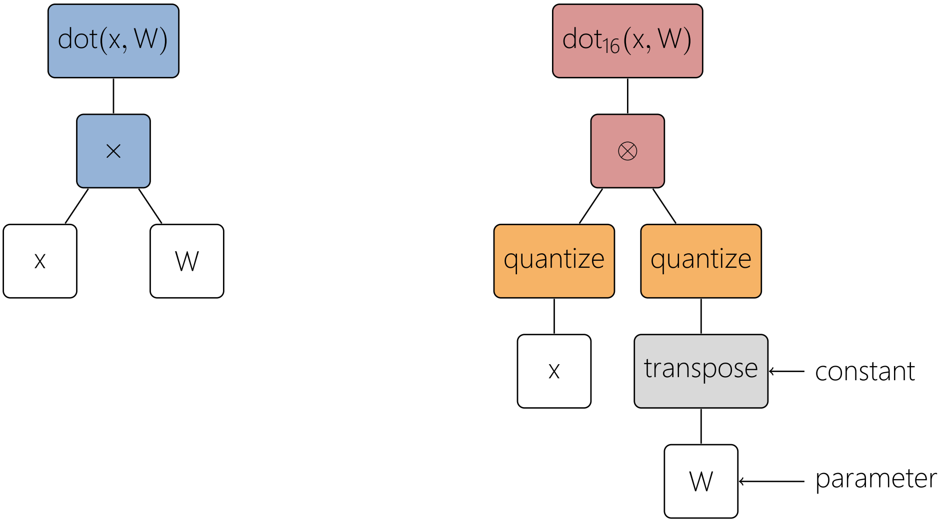

$\mathrm{dot}_{16}(x, W)$
⊗
quantize
quantize
x
transpose
W

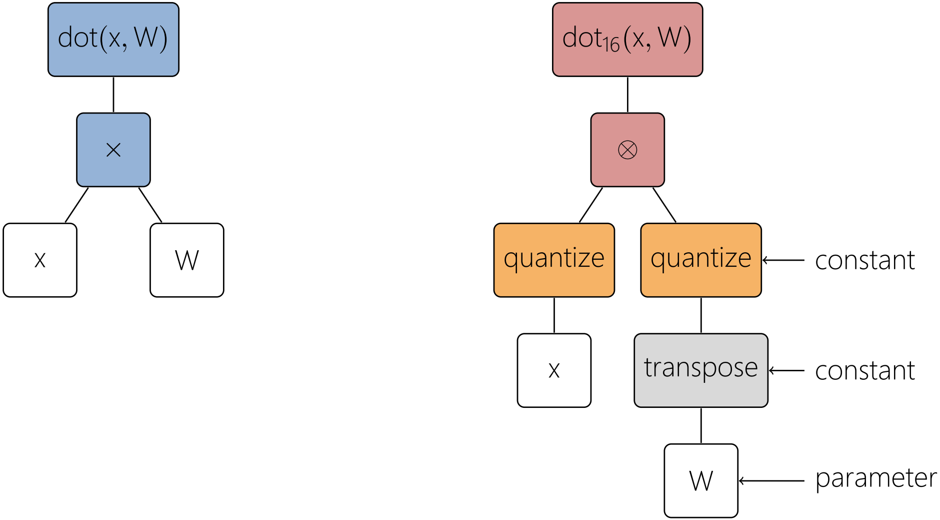

MM-MKL
MM-int16
Quant-int16
Other
Transformer-base
+Shortlist
+AAN (modified)
+MM-int16
+Memoization
+Auto-tuning
+Caching att. keys/values
w/o --optimize
542.6
412.0
304.7
505.9
0
100
200
300
400
500
600
Seconds to translate newstest2014 (batch-size: ca. 384 words – 5 to 25 sentences)

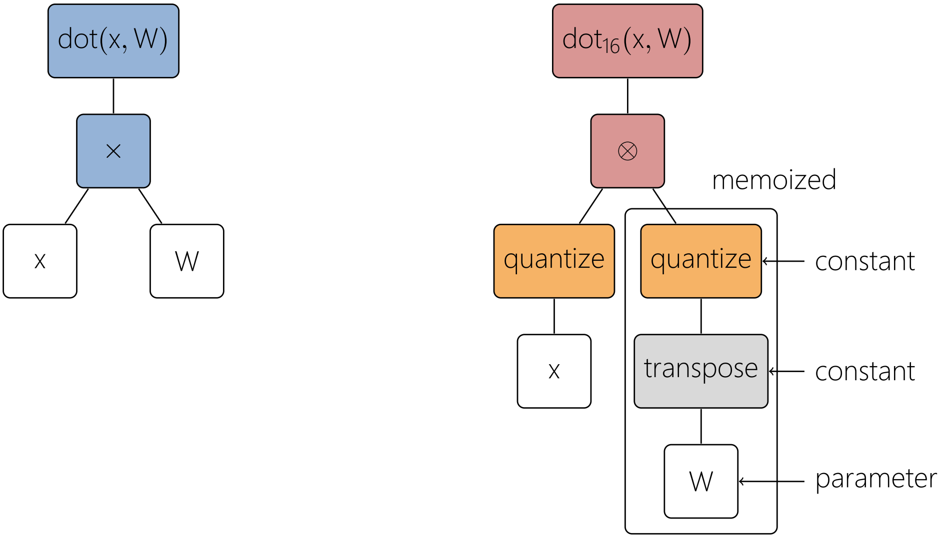

dot(x, W)
×
x
W
dot16(x, W)
⊗
quantize
quantize
x
transpose
W



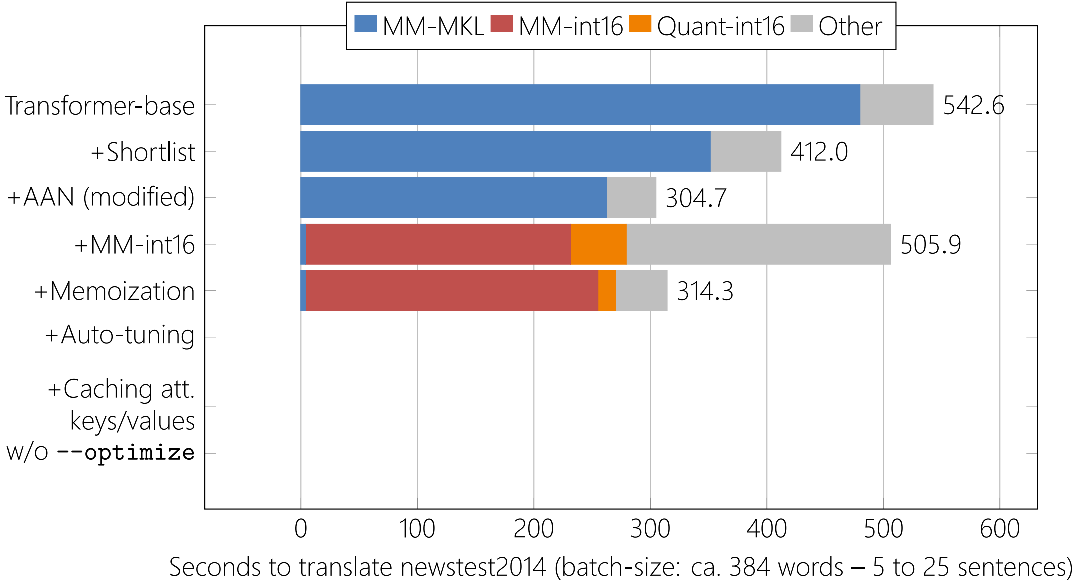
dot(x, W)
×
x
W
dot16(x, W)
⊗
quantize
quantize
x
transpose
W
parameter

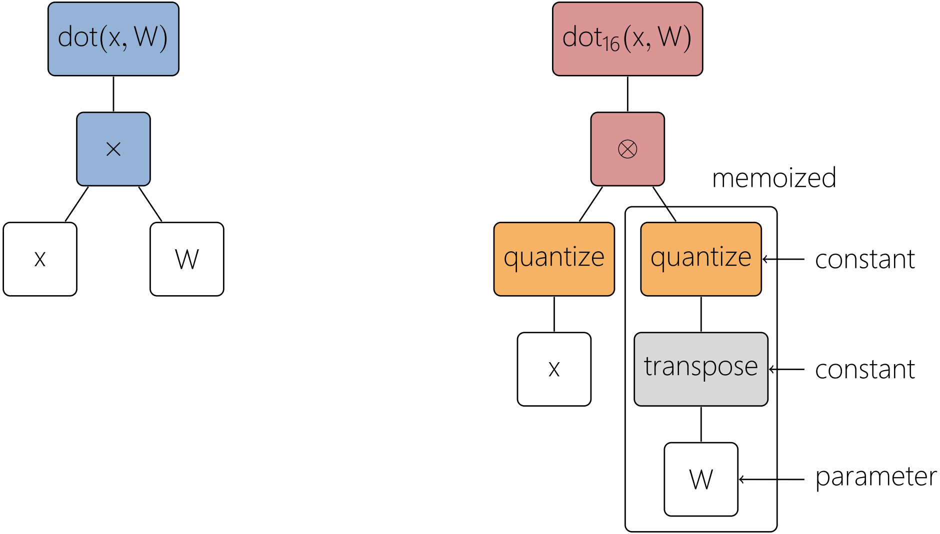

dot(x, W)
×
x
W
dot16(x, W)
⊗
quantize
quantize
x
transpose
constant
W
parameter

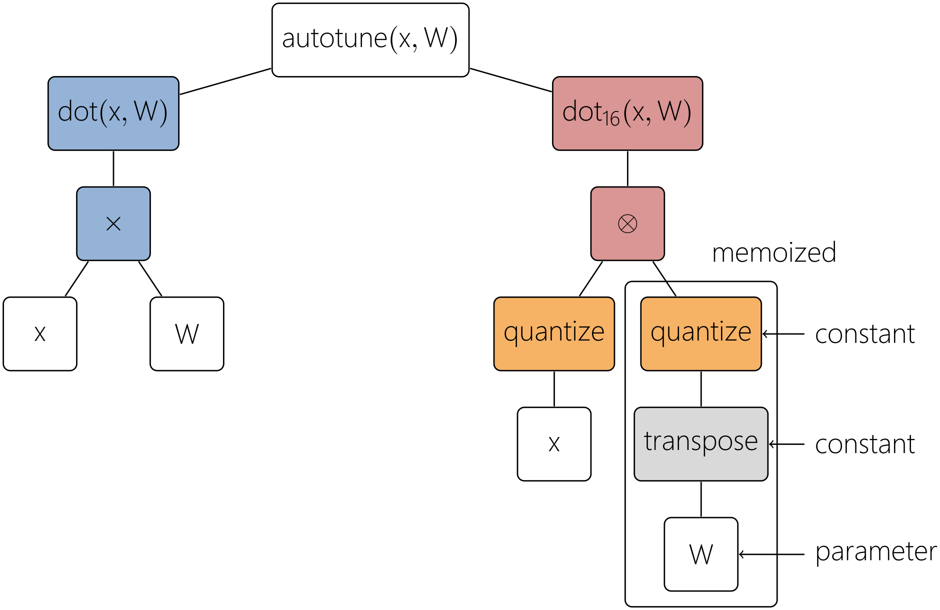

dot(x, W)
×
x
W
dot16(x, W)
⊗
quantize
quantize
constant
x
transpose
constant
W
parameter



dot(x, W)
×
x
W
dot16(x, W)
⊗
memoized
quantize
quantize
constant
x
transpose
constant
W
parameter


MM-MKL
MM-int16
Quant-int16
Other
Transformer-base
+Shortlist
+AAN (modified)
+MM-int16
+Memoization
+Auto-tuning
+Caching att. keys/values
w/o --optimize
542.6
412.0
304.7
505.9
314.3
0
100
200
300
400
500
600
Seconds to translate newstest2014 (batch-size: ca. 384 words – 5 to 25 sentences)

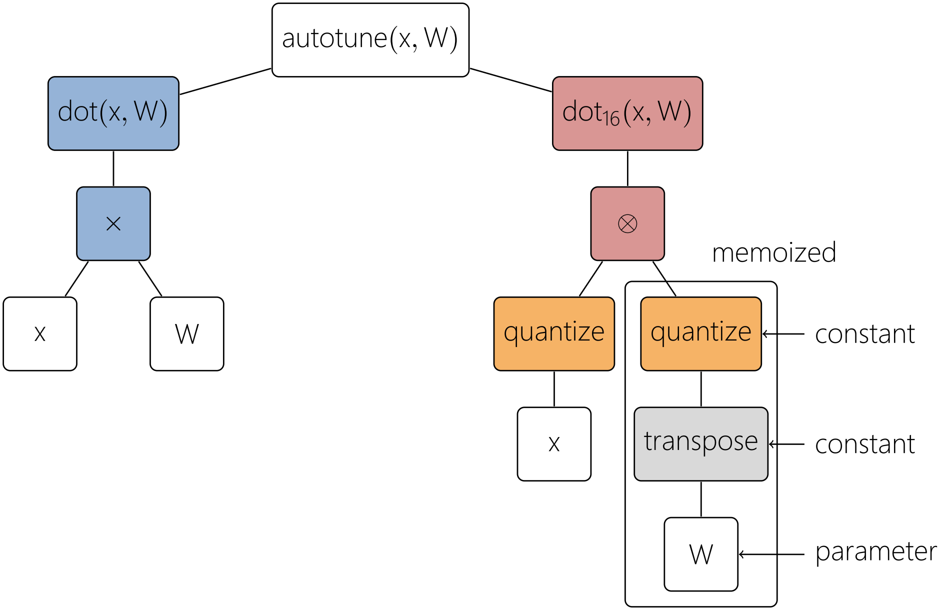

dot(x, W)
×
x
W
dot16(x, W)
⊗
memoized
quantize
quantize
constant
x
transpose
constant
W
parameter

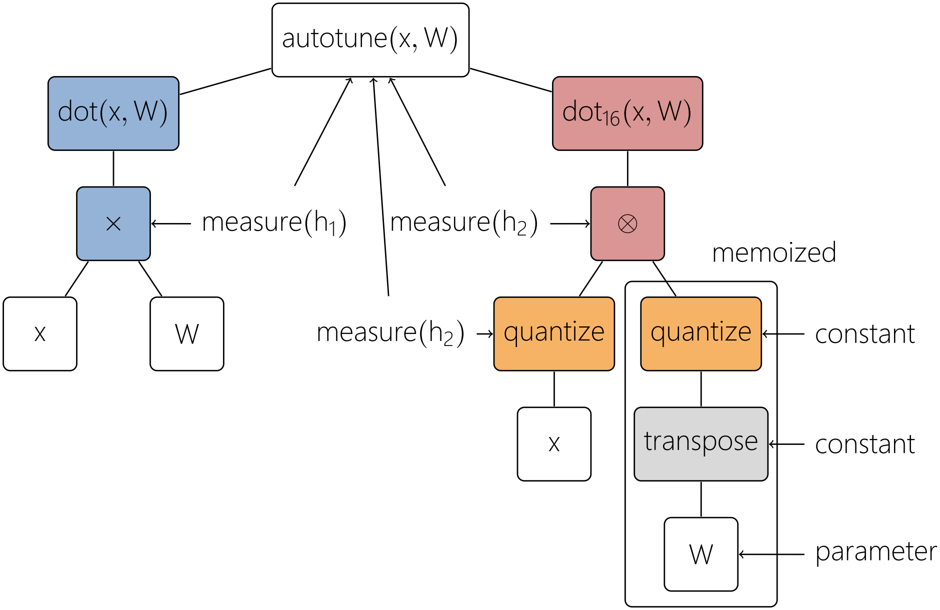

autotune(x, W)
dot(x, W)
×
x
W
dot16(x, W)
⊗
memoized
quantize
quantize
constant
x
transpose
constant
W
parameter

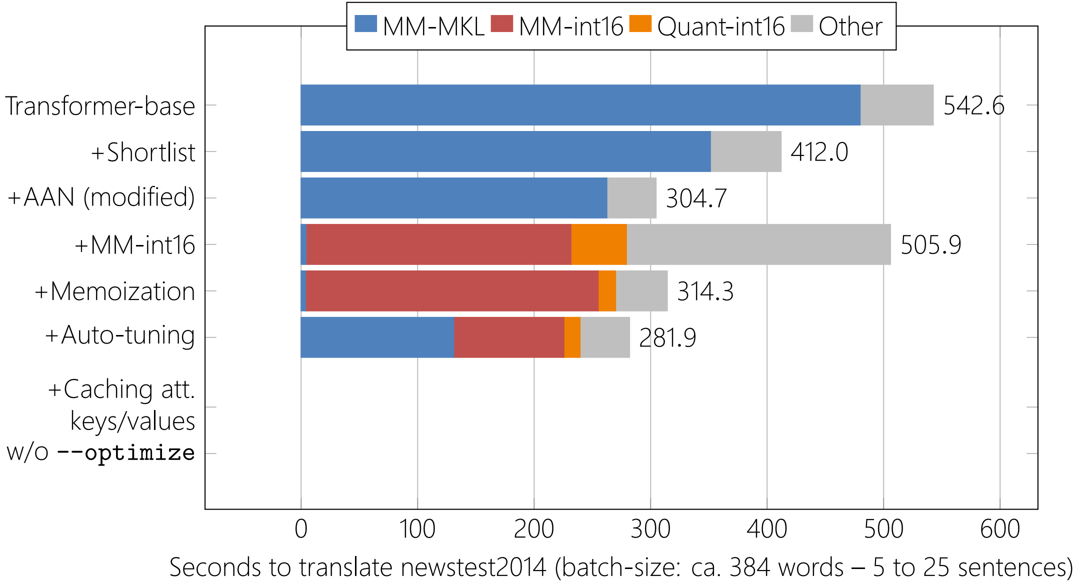
$$
\begin{aligned}
h_1 &= \mathrm{hash}(\mathrm{dot}(x, W)) \\
&= \mathrm{hash}(\mathrm{dot}) \odot \mathrm{hash}(\mathrm{dims}(x)) \odot \mathrm{hash}(\mathrm{dims}(W)) \\
&= \mathrm{hash}(\mathrm{dot}) \odot \mathrm{hash}(\{11, 512\}) \odot \mathrm{hash}(\{512, 512\})
\end{aligned}
$$

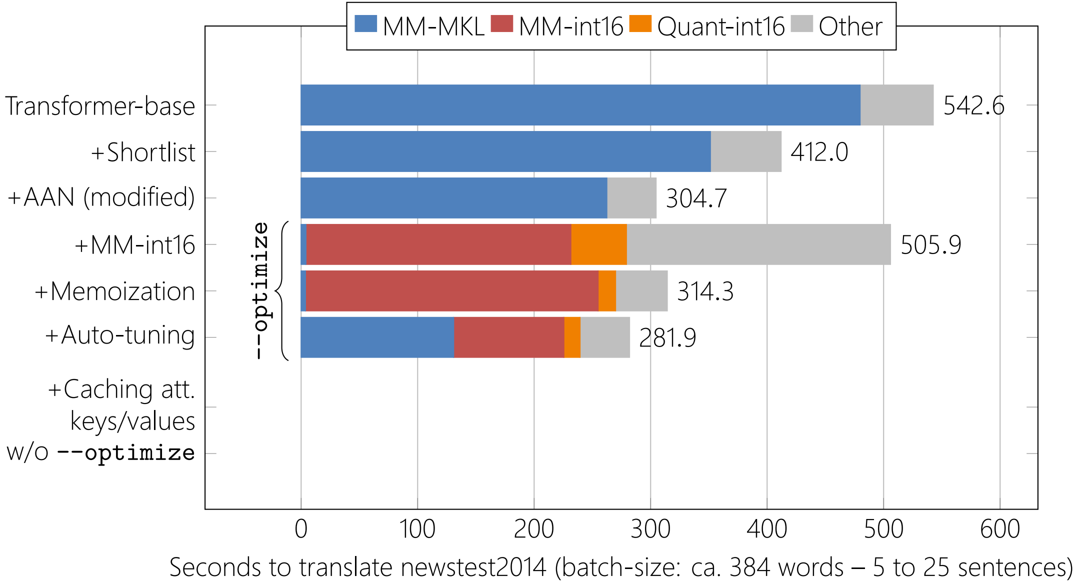
$$
\begin{aligned}
h_2 &= \mathrm{hash}(\mathrm{dot}_{16}(x, W)) \\
&= \mathrm{hash}(\mathrm{dot}_{16}) \odot \mathrm{hash}(\mathrm{dims}(x)) \odot \mathrm{hash}(\mathrm{dims}(W)) \\
&= \mathrm{hash}(\mathrm{dot}_{16}) \odot \mathrm{hash}(\{11, 512\}) \odot \mathrm{hash}(\{512, 512\})
\end{aligned}
$$

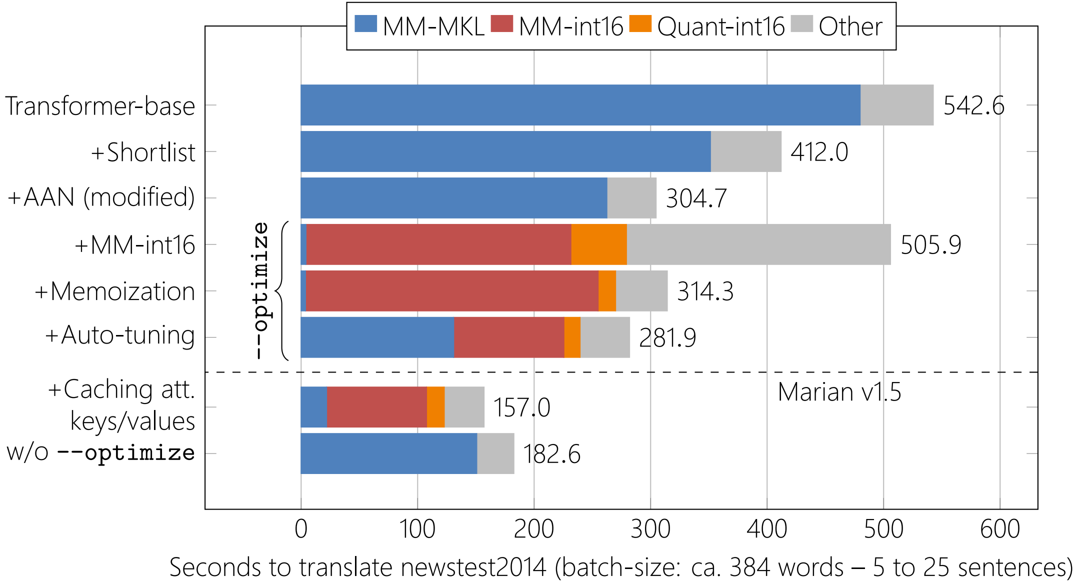
We can decrease granularity via integer-dividing dimensions by a given factor, we choose 5.


autotune(x, W)
dot(x, W)
×
x
W
dot16(x, W)
⊗
memoized
quantize
quantize
constant
x
transpose
constant
W
parameter

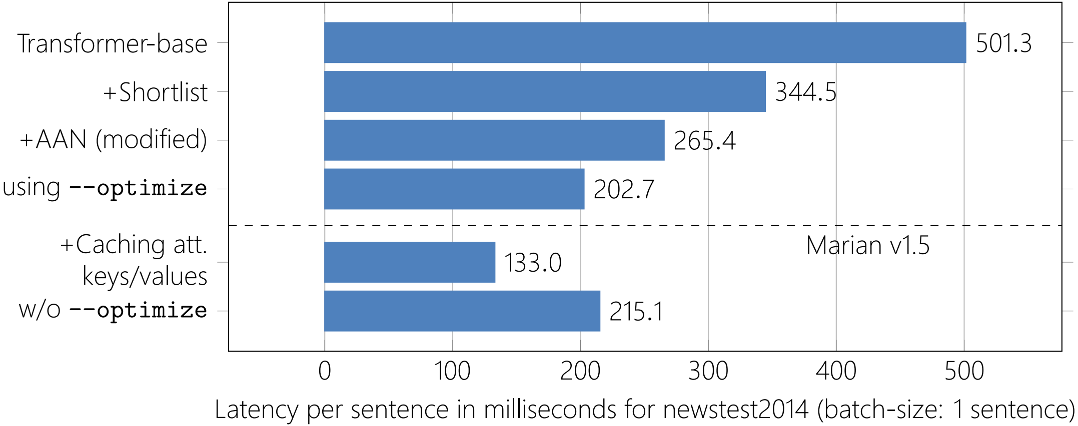

autotune(x, W)
dot(x, W)
dot16(x, W)
×
measure(h1)
measure(h2)
⊗
memoized
x
W
measure(h2)
quantize
quantize
constant
x
transpose
constant
W
parameter


MM-MKL
MM-int16
Quant-int16
Other
Transformer-base
+Shortlist
+AAN (modified)
+MM-int16
+Memoization
+Auto-tuning
+Caching att. keys/values
w/o --optimize
542.6
412.0
304.7
505.9
314.3
281.9
0
100
200
300
400
500
600
Seconds to translate newstest2014 (batch-size: ca. 384 words – 5 to 25 sentences)


MM-MKL
MM-int16
Quant-int16
Other
Transformer-base
+Shortlist
+AAN (modified)
+MM-int16
+Memoization
+Auto-tuning
+Caching att. keys/values
w/o --optimize
--optimize
542.6
412.0
304.7
505.9
314.3
281.9
0
100
200
300
400
500
600
Seconds to translate newstest2014 (batch-size: ca. 384 words – 5 to 25 sentences)



MM-MKL
MM-int16
Quant-int16
Other
Transformer-base
+Shortlist
+AAN (modified)
+MM-int16
+Memoization
+Auto-tuning
+Caching att. keys/values
w/o --optimize
--optimize
542.6
412.0
304.7
505.9
314.3
281.9
157.0
182.6
Marian v1.5
0
100
200
300
400
500
600
Seconds to translate newstest2014 (batch-size: ca. 384 words – 5 to 25 sentences)

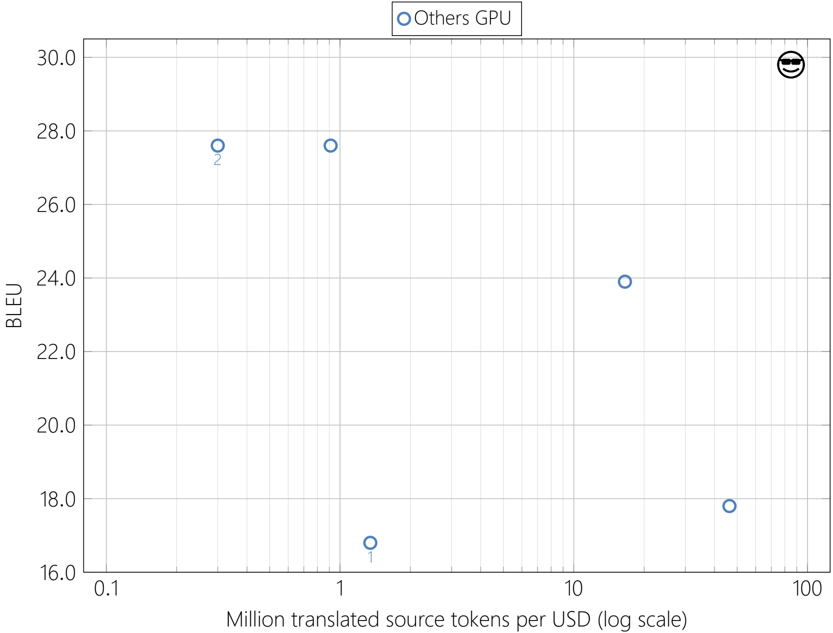

Transformer-base
+Shortlist
+AAN (modified)
using --optimize
+Caching att. keys/values
w/o --optimize
501.3
344.5
265.4
202.7
133.0
215.1
Marian v1.5
0
100
200
300
400
500
Latency per sentence in milliseconds for newstest2014 (batch-size: 1 sentence)

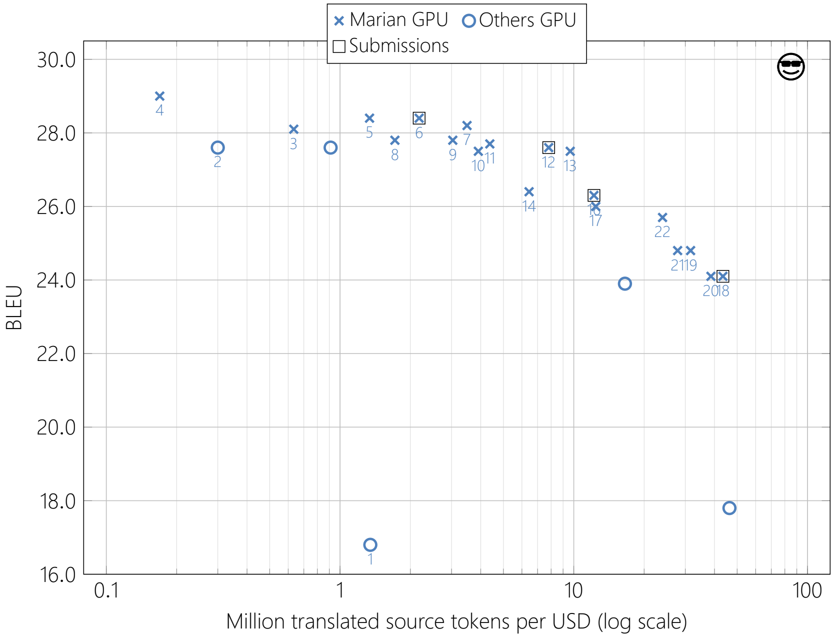
Map GPU and CPU performance into comparable space [w/$]

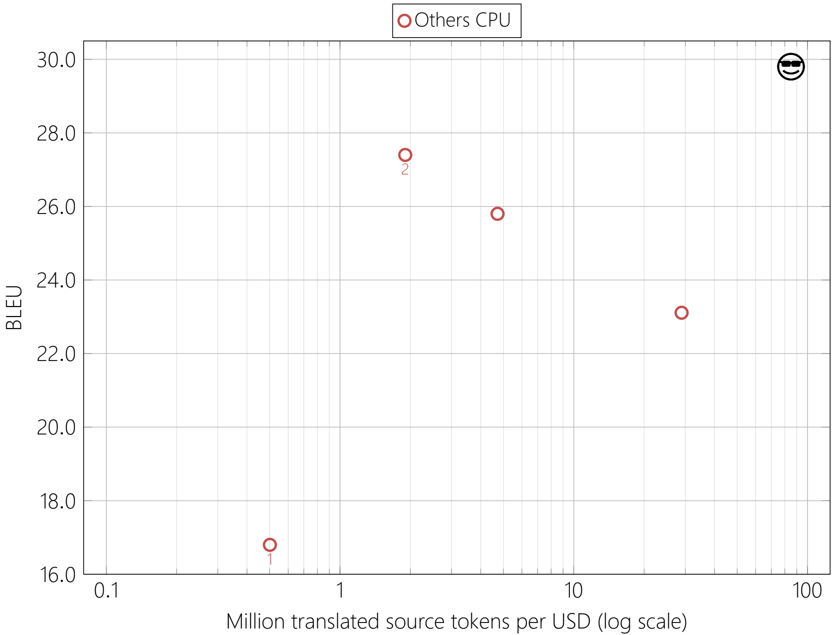
newstest2014.de consists of 62,954 tokens

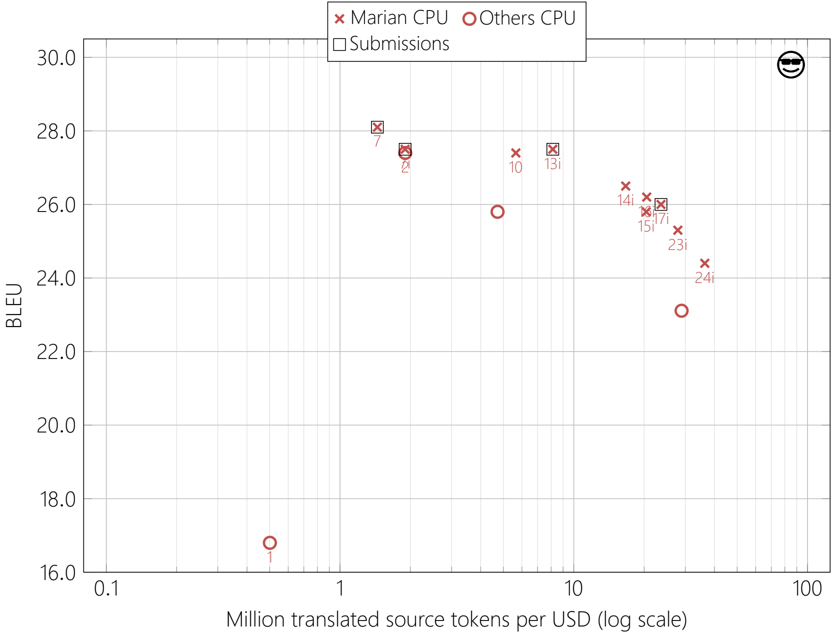
Type	Price [$/h]
p3.x2large	3.259
m5.large	0.102

$$[\mathrm{w}/\$] = \frac{62,954\ [\mathrm{w}]}{\text{Translation time [s]}} \cdot \frac{3,600\ [\mathrm{s/h}]}{\text{Instance price } [\$/\mathrm{h}]}.$$

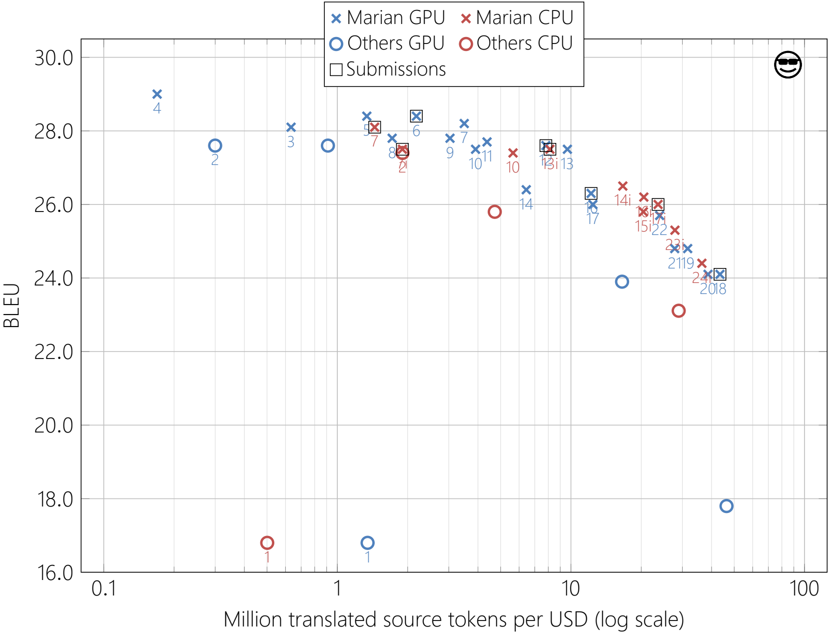

Others GPU
BLEU
30.0
28.0
26.0
24.0
22.0
20.0
18.0
16.0
2
1
0.1
1
10
100
Million translated source tokens per USD (log scale)

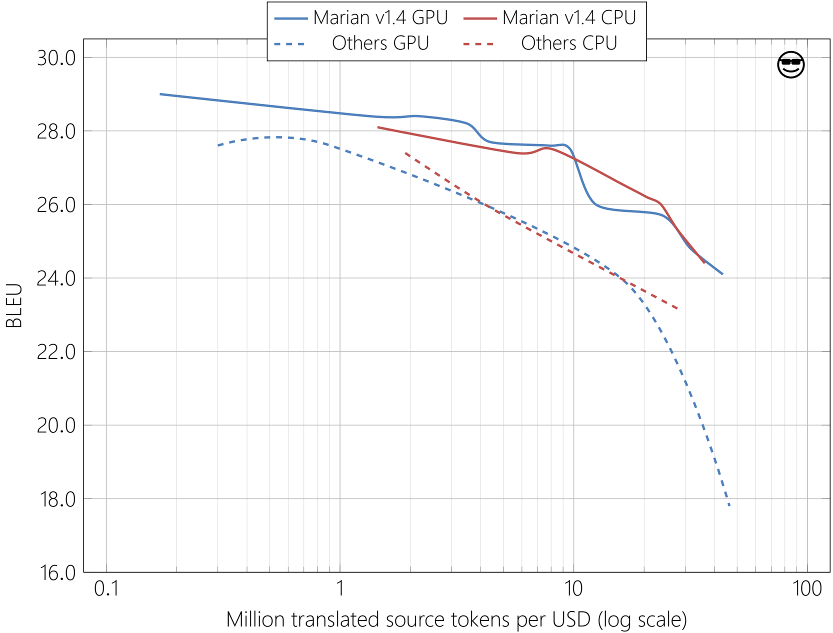

Marian GPU
Others GPU
Submissions
30.0
28.0
26.0
24.0
22.0
20.0
18.0
16.0
BLEU
0.1
1
10
100
Million translated source tokens per USD (log scale)
1
2
3
4
5
6
7
8
9
10
11
12
13
14
17
18
19
20
21
22

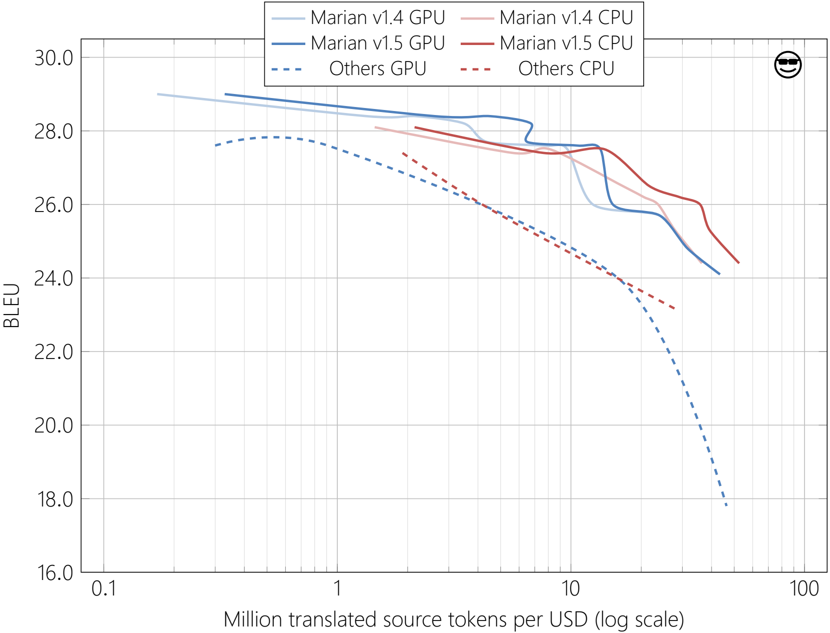

Others CPU
30.0
28.0
26.0
24.0
22.0
20.0
18.0
16.0
BLEU
2
1
0.1
1
10
100
Million translated source tokens per USD (log scale)



Marian CPU
Others CPU
Submissions
BLEU
Million translated source tokens per USD (log scale)
30.0
28.0
26.0
24.0
22.0
20.0
18.0
16.0
0.1
1
10
100
7
2
10
13i
14i
15i
16i
17i
23i
24i
1


Marian GPU
Marian CPU
Others GPU
Others CPU
Submissions
BLEU
30.0
28.0
26.0
24.0
22.0
20.0
18.0
16.0
0.1
1
10
100
Million translated source tokens per USD (log scale)


Marian v1.4 GPU
Marian v1.4 CPU
Others GPU
Others CPU
30.0
28.0
26.0
24.0
22.0
20.0
18.0
16.0
BLEU
0.1
1
10
100
Million translated source tokens per USD (log scale)



Marian v1.4 GPU
Marian v1.4 CPU
Marian v1.5 GPU
Marian v1.5 CPU
Others GPU
Others CPU
30.0
28.0
26.0
24.0
22.0
20.0
18.0
16.0
BLEU
0.1
1
10
100
Million translated source tokens per USD (log scale)

Future work

- More experiments with Teacher-Student scenario;
- More SIMD operations on the CPU;
- All operations in fixed-point arithmetics on the CPU;
- 8-bit matrix product on the CPU;
- Mixed precision (FP16) on the GPU;
- Optimize beam-search for batched translation;
- ...

Microsoft

We are hiring!

Talk to me if you are interested.

Translator