USC Viterbi School of Engineering
Information Sciences Institute

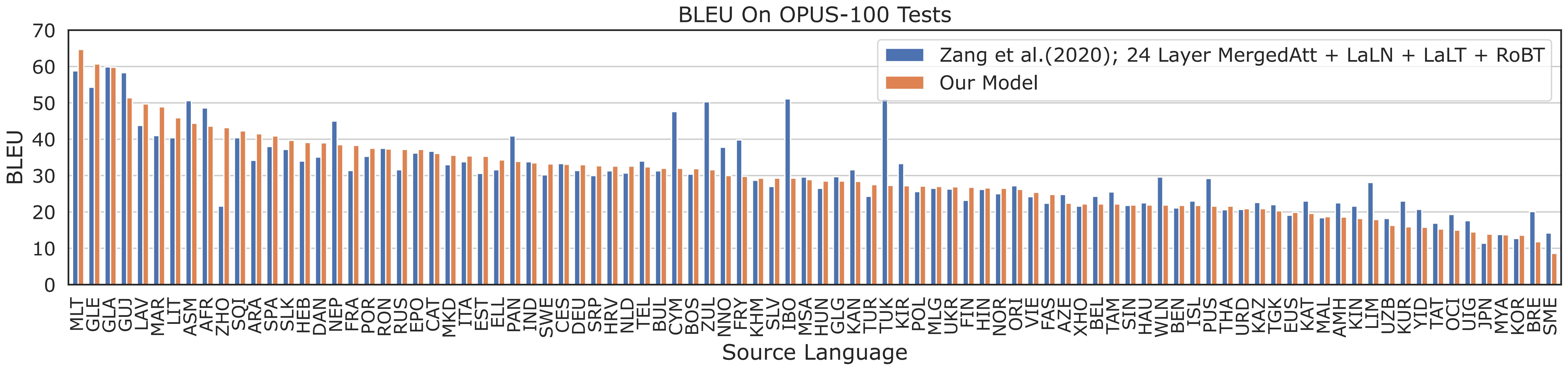
Many-to-English Machine Translation Tools, Data, and Pretrained Models

Thamme Gowda Zhao Zhang Chris Mattmann Jonathan May

Introducing three tools for machine translation; reproducibility and scalability in focus

- *MTData*: dataset catalogue and downloader
- *NLCodec*: vocabulary manager, dataset storage and retrieval layer
- *RTG*: NMT toolkit, based on PyTorch; from research to production

- Massive dataset: 500+ languages, 474 million sentences, 9 billion tokens each side
- Massively multilingual NMT model: 500 languages → English translation
- Home page: http://rtg.isi.edu/many-eng



MTData

```
pip install mtdata
```

- Reproducible experiments: clear way of communicating MT dataset
- Index of publicly available parallel datasets (120K+ as of June 2021)
- Maps language names to ISO-639-3
- Unified interface to datasets from heterogeneous sources, and formats
- Hides mundane tasks, e.g., locating URLs, downloading, decompression, parsing, and sanity checking
- Parses heterogeneous data formats for parallel datasets and produces plain text files after merging
- Reduces network transfers by maintaining a local cache, which is shared between experiments
- Sanity checks such as segment count matching
- Shows BibTeX attributed to datasets
- github.com/thammegowda/mtdata

NLCodec

```
pip install nlcodec
```

- Flexible and scalable vocabulary manager and storage layer
- Flexible options: word, char, BPE
- Uses PySpark backend for scaling

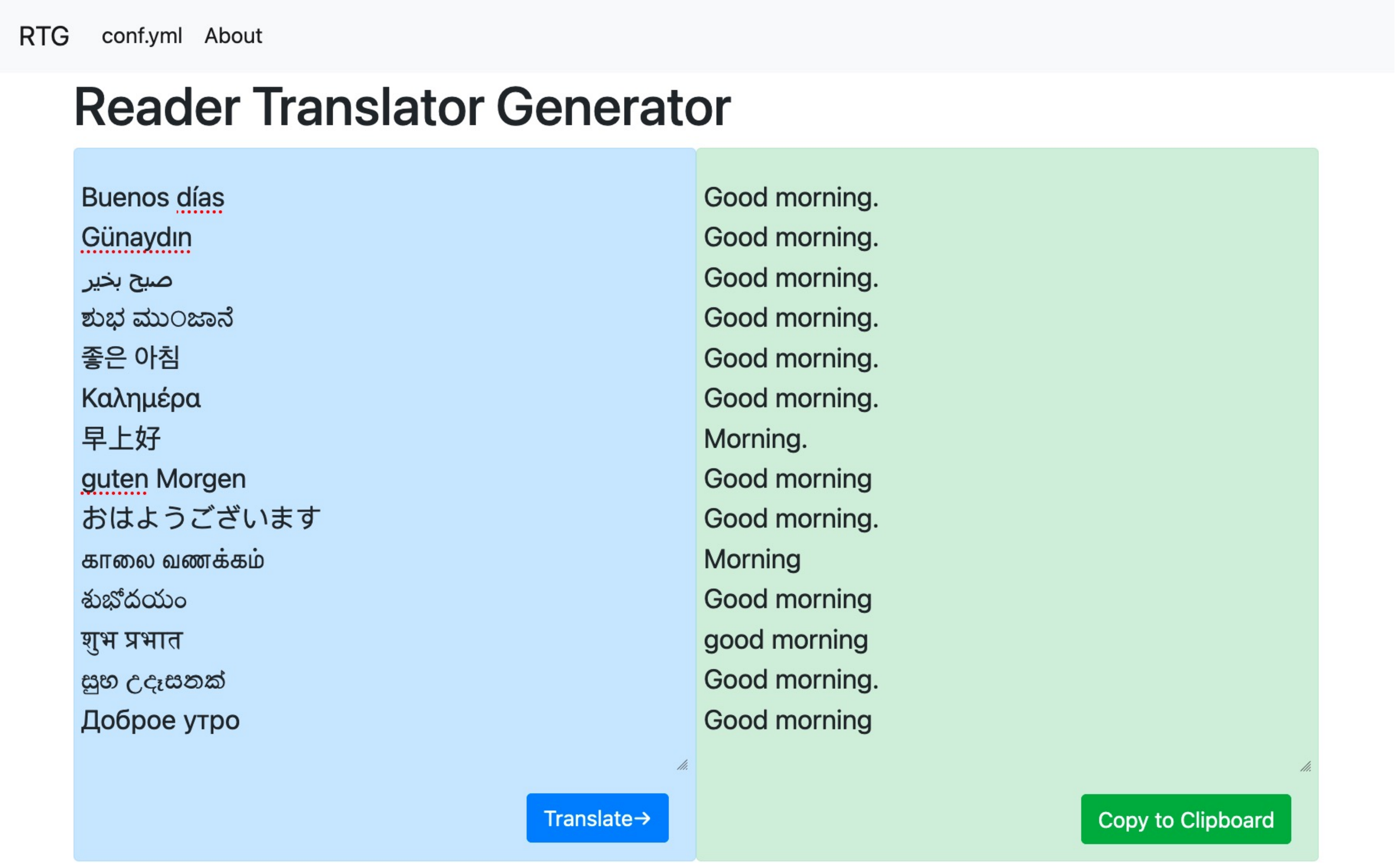
NLDb

- Efficient storage layer; adapts integer byte size based on vocabulary size
- Memory efficient by adapting datatypes based on vocabulary size.
- 1-byte unsigned int for vocabulary size < 256 (Latin chars)
- 2-byte unsigned int for BPE vocabs up to 65,536 types, 4-byte for the rest
- Offers a multi-part database with horizontal sharding; supports *parallel writes* (e.g., Apache Spark) and *parallel reads* (distributed training).
- Batching such as random batches with approx-equal-length sequences
- github.com/isi-nlp/nlcodec

RTG

```
pip install rtg
```

- Reader translator generator (RTG) is a NMT toolkit based on Pytorch
- Reproducible experiments; all the required parameters of an experiment are included in a single YAML config file; can be shared easily.
- Transformers, and RNN with x-attn
- Supports distributed training on multi-node multi-GPUs, gradient accumulation, Float16 operations, and integrated Tensorboard
- Tied embedding, parent-child transfer, beam decoding with length normalization, early stopping, and checkpoint averaging
- Flexible vocabulary options with NLCodec and SentencePiece which can be either shared or separated between source and target languages
- CLI, REST API and Web UI
- isi-nlp.github.io/rtg



Reader Translator Generator

Buenos días	Good morning.
Günaydın	Good morning.
صبح بخير	Good morning.
ಶುಭ ಮುಂಜಾನೆ	Good morning.
좋은 아침	Good morning.
Καλημέρα	Good morning.
早上好	Morning.
guten Morgen	Good morning
おはようございます	Good morning.
காலை வணக்கம்	Morning
శుభోదయం	Good morning
शुभ प्रभात	good morning
සුභ උදෑසනක්	Good morning.
Доброе утро	Good morning

Usage

```
$ mtdata list -l <l1-l2>
$ mtdata get -l <l1-l2> -tr <train> --merge -tt <test> -o <out>
$ nlcodec [learn|encode|decode] -m <model>
$ rtg-pipe <experiment/dir>

$ IMAGE=tgowda/rtg-model:500toEng-v1
$ docker run --gpus '"device=0"' --rm -i -p 6060:6060 $IMAGE
$ curl --data “source=<text>”  http://localhost:6060/translate
```