

MT Marathon 2018 (Prague)

Proposed Projects

Start by **sharing the document with yourself** (so that Google records edits well).

Then simply start a new entry or add your comments anywhere, in the text or on side.

Projects can be proposed until the first day of MT Marathon, but announcing them earlier might attract more participants, come better prepared etc.

List of Projects (automatically generated)

[Sample Project](#)

Sample Project

(John the Proposer; include e-mail if you want to [by doing so, you indicate you understand all the implications of GDPR, so I would never dare to ;-])

- The bright idea.
- The goal.
- Prospective participants: John the Proposer

Bootstrap resampling for other metrics

Mārcis Pinnis <marcis.pinnis@tilde.lv>, Matīss Rikters <matiss.rikters@tilde.lv>

Multeval (<https://github.com/jhclark/multeval>) currently has bootstrap resampling for BLEU, METEOR and TER. It would be useful to add more recent metrics like ChrF3, CharactER and BEER 2.0 in Multeval or some other evaluation toolkit (could be also based on SacreBLEU...).

- Ondrej Bojar: Great! Ideally, include Moses Scorer (C++, aka [evaluator](#)), which has chrF3 in, aside from BLEU, PER, WER, CDER and perhaps TER.

Self-adaptive NMT for Sockeye or Marian

Mārcis Pinnis <marcis.pinnis@tilde.lv>, Rihards Krišlauks <rihards.krislauks@tilde.lv>

The idea would be to implement the online adaptation feature from Modern MT (<https://github.com/ModernMT/MMT>) in Sockeye or Marian and test it on Transformer models.

- Roman Grundkiewicz: Could you provide a description how does the self-adaptation feature work? I have implemented a working proof of concept of the method from FBK (<http://www.statmt.org/wmt17/pdf/WMT13.pdf>) in Marian and could be interested in improving and releasing it.
- Mārcis Pinnis: Great! This is exactly what we were thinking of implementing 🕶️ (the MMT toolkit was created in a project where FBK was involved)

Neural Monkey Unsupervised

Ondřej Bojar <bojar@ufal.mff.cuni.cz>, Ivana Kvapilíková <ivana.kvapilikova@gmail.com>

The goal of the project would be to create a toolkit for unsupervised machine translation. We

would like to set up unsupervised training for Neural Monkey (<https://github.com/ufal/neuralmonkey>) to be able to learn without using parallel corpora. Unsupervised systems are also represented in this year's [WMT18](#). The following papers are dealing with this topic: [link1](#), [link2](#), [link3](#), [link4](#).

Variational auto-encoders for dependency trees

Jindřich Helcl <helcl@ufal.mff.cuni.cz>, Dušan Variš <varis@ufal.mff.cuni.cz>

Explicit syntax features can help NMT performance, however require additional inference steps in an otherwise end-to-end solution. We propose a regularization technique that trains the encoder to predict a dense syntax representation of the source sentence in a multi-task learning scenario. For this, we propose to obtain dense representations of syntactic trees using variational auto-encoders on large treebanks and then applying them on parsed parallel data.

Multilingual MT

Aleš Tamchyna <ales.tamchyna@memsource.com>

Multilingual models are slowly approaching a level of quality suitable for production use. The goal of this project would be to use some of the recent papers and implement a full, usable multilingual system, including a simple and robust training pipeline. Possible sources of ideas:

- neural interlingua: <https://arxiv.org/abs/1804.08198>
- dual zero-shot translation: <https://arxiv.org/abs/1805.10338>

Zero-shot Quality Estimation

Aleš Tamchyna <ales.tamchyna@memsource.com>

MT QE is a task closely related to machine translation. The goal of QE models is to predict the quality of MT output given the source. In this project, we would explore whether it's possible to reuse work on multilingual MT models to obtain a multilingual, and ideally zero-shot, quality estimation model. Besides techniques for multilingual MT, we could look at approaches from cross-domain or cross-lingual text classification where domain-adversarial training seems to work well.

Parallel Corpus Mining with Deep Learning

Hieu Hoang <hieuhoang@gmail.com>

Extracting parallel corpus from the web is an interesting and useful task in order to create good quality MT systems. A related task, parallel corpus filtering, was a shared task at this year's WMT conference.

However, parallel corpus mining is not a subject that attract much research attention and current methods can be improved. Recently, there has been some research in this area such as the following papers

- <http://www.iro.umontreal.ca/~felipe/bib2webV0.81/cv/papers/paper-francis-coling-2018.pdf>
- <https://arxiv.org/abs/1807.11906>

This MTM project will be a starting block for those who want to employ deep learning to a new field which will hopefully lead to some publishable research. We will re-implement and extend one of the paper mentioned

Neural interlingua for MT

Raul Vazquez <raul.vazquez@helsinki.fi>, Alessandro Raganato <alessandro.raganato@helsinki.fi>

Building a modular framework for multilingual MT, using double-attentive models with language-specific encoders and decoders that can be efficiently trained with a language-rotating schedule.

- **Idea:** Implement a compound attention architecture based on the one proposed by [Cifka and Bojar \(2018\)](#) in a multilingual setup. For this, a training scheduler will be needed (see [Lu, et al., 2018](#)).
- **Goal:** get a fully functional implementation of the proposed framework in OpenNMT-py, such that the learned weights of any of the layers can be easily extracted and tested in downstream tasks ([SentEval](#)).

Highly Scalable Neural MT for Production Environments with Marian

Ulrich Germann <ugermann@inf.ed.ac.uk>

Most Neural MT toolkits are geared towards batch processing from files, which is OK for experimentation and WMT-style shared tasks, but suboptimal for deployment as a back-end service in production environments. This project aims at adding an [AMQP](#) worker interface to Marian, so that large workloads can be spread dynamically across multiple instances of Marian running on multiple hosts, at the same time facilitating continuous uptime of MT services even when specific machines need to go down for maintenance or other reasons occasionally. We will be using [RabbitMQ](#) as the AMQP message broker, and probably [AMQP-CPP](#) as the C++ amqp library. Time permitting, we will also implement a light-weight REST API front-end for a generic translation service. As a starting point, we can define the API within the [OpenAPI](#) framework and use [Swagger Tools](#) to generate the stubs for a light-weight web server front end that will push translation requests onto the message queue and assembles responses from it. This project builds upon work done in the [SUMMA](#) project, but there we currently have a python wrapper that talks to Marian via the current lightweight websocket implementation, which is not designed for heavy loads. Prowess in Python (web front-end; ideally with experience in asynchronous Python > 3.6 programming) or C++ (back-end) is required for participation in this project.

NMT Inspector

Jan Niehues (jan.niehues@kit.edu)

Currently, neural machine translation systems are mainly treated as a black box. In order to better understand the behavior of NMT, the project should develop a toolkit that helps to inspect hidden representation of NMT models. Based on an initial toolkit (<https://github.com/jniehues-kit/NMTInspector>), new methods to get insides into the models as well as the support of other NMT toolkits should be implemented. Therefore, different machine learning techniques will be explored: Classification, Clustering, ...

MT-CompareEval Improvements

MT-CompareEval is a very useful tool for comparison and evaluation of machine translations. Despite it's used by many users, it suffers on several issues, from which one of the most criticals is the low speed of importing new translation tasks. A possible remedy can be e.g. improvement of the current PHP module, reimplementing of the task importer in Python, reimplementing of the whole front-end and back-end in any faster toolkit etc. Work on this project probably requires software engineering in any web framework or scripting language, understanding of the current PHP code, consultations with original authors etc.

Links:

- Live demo: <http://mt-compareval.ufal.cz/>
- Original paper: Ondřej Klejch, Eleftherios Avramidis, Aljoscha Burchardt, Martin Popel: [MT-CompareEval: Graphical evaluation interface for Machine Translation development](#). The Prague Bulletin of Mathematical Linguistics, No. 104, 2015, pp. 63–74.
- GitHub: <https://github.com/choko/MT-CompareEval>
- Issue with speed: <https://github.com/choko/MT-CompareEval/issues/80>

This project is proposed by Dominik Macháček <gldkslfmsd@gmail.com>, but he doesn't intend to participate on this project on MTM18 :(.

Universal Transformers in Sockeye

David Vilar <dvilar@amazon.com>

Recently the universal transformer model has been introduced by Dehghani et al. This model modifies the baseline transformer model by treating the depth of the model as a recursive dimension. With this modification the model becomes computationally universal. The goal of this project is to reimplement this model within the sockeye toolkit. The original paper also gave mixed results for machine translation. Although the authors report improvements, they had to restrict the expressiveness of the model. It would be beneficial to better understand this behaviour of the system for MT.

Hierarchical Decoding for Character-Level NMT

Duygu Ataman, ataman@fbk.eu

Different architectures for character-level NMT have been proposed, which generally model character generation at the sentence level. However, both LSTM and Transformer based NMT models suffer when learning long-range dependencies. In this project, we will implement hierarchical decoding which propagates through both word and character levels during translation, allowing the target language model to better capture semantic and morphological features. The project is going to be implemented in OpenNMT-py using pytorch, and the main aim will be exploring different strategies for input feeding and architecture choices.