MT Marathon 2019 (Edinburgh)

Project proposals

Feel free to 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.

Sample project

(John the Proposer <your.email@if.you.want.to>)

- The bright idea.
- The goal.
- Prospective participants.

Visualization tool for Marian

(Roman Grundkiewicz)
TensorBoard provides the visualization and tooling of ML experiments for TensorFlow. The idea is to add support to a stand-alone TensorBoard for Marian, using the information generated by Marian into log files during training. An attempt to extract TensorBoard from TensorFlow has been made for MXNET (mxconsole), and there is TensorBoardX for PyTorch. An interesting alternative is MLFlow. The goal of the project would be to visualize curves for training and validation metrics. Requirements: basic Python

Diverse Translations With Sentence Codes

(Matt Post)
Many times we'd like diverse outputs from NMT systems. Existing approaches include beam search, constrained decoding, or sampling, but there are problems with all of these: beam search doesn't produce very diverse outputs, constrained decoding is slow and requires a source of constraints, and sampling can introduce inadequate garbage tokens. At ACL this year, Shu et al. proposed to prefix target sentences with "sentence codes" (https://www.aclweb.org/anthology/P19-1177/), which are learned from applying an auto-encoder to parse trees produced over the target side of the training data. This allows one to get syntactically or semantically diverse outputs by force-decoding with one of the codes—selected at random from the full set—as a prefix. In this project, we will reimplement their work.

The tasks will then be to:

- Write the autoencoder, preferably in pytorch
- Apply it to the parse trees
- Cluster the codes
- Train a new translation system and evaluate it

We will open source all code.

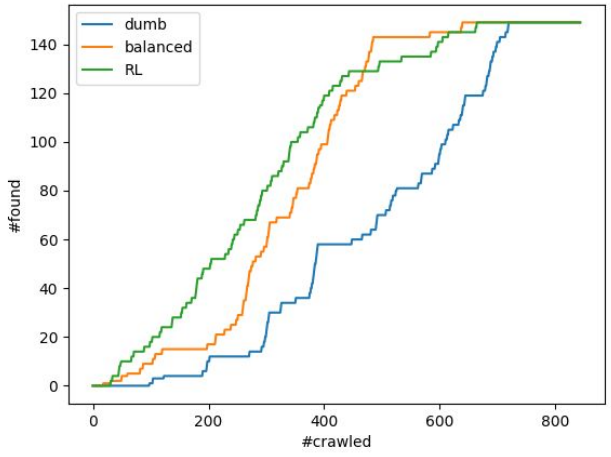
Intelligent Crawling for Parallel Data

(Hieu Hoang)

Multilingual websites are a useful source of parallel data. However, exhaustive crawling and processing of many large websites is infeasible due to the number of pages they contain. This project will use reinforcement learning (RL) to develop strategies in these cases where we must make the risky decisions on which links to follow and pages to crawl, and what to leave out.

Current open-source parallel corpus collection toolkits (eg. Bitextor) use off-the-shelf crawlers to download webpages which typically aim to mirror the website by traversing pages methodically depth-first or breadth first. The crawler has no knowledge of the downstream need to find matching webpages in multiple languages, or even the language of the pages. This reduces the efficiency of the pipeline and number of parallel pages found.

Other ways of traversing websites can result in more parallel pages found faster. For example, an algorithm which seeks to crawl an equal number of pages from both languages of interest (‘balanced’) significantly outperform the off-the-shelf crawler (‘dumb’), see Figure below. We’ve also started on RL-based methods (DQN and REINFORCE) to improve on this ‘balanced’ baseline. We aim to continue this line of research.



This project will be of interest to anyone who is interested in, or has expertise in, RL, deep learning, graph representation (eg. gated graph network) or wants to use it to create parallel corpora.

MTComparEval for large-scale experiments and transfer learning

(CrossLang)

MTComparEval (https://github.com/choko/MT-ComparEval) is a nice tool for plotting the performance of MT systems and comparing individual translations for a limited number of test sets. If, however, you want to track the performance of your systems, grouped per language pair, over a larger number of test sets, MTComparEval tends to get cluttered.

We thought it was a nice idea to continue the work on MTComparEval and add the following features:

- Grouping per language pair
- Grouping of multiple test sets per language pair
- Exporting a selection of translated sentences (typically interesting low-quality translations) for post-editing
- A hierarchical view for visualising provenance of MT systems trained using transfer learning
- Switching to a faster database

We asked Ondrej Klejch (https://github.com/choko) how he felt about this, and we have his full support!

You can find our fork here: https://github.com/CrossLangNV/MT-ComparEval and some screenshots below. **We commit to keeping everything Open Source under Apache 2.0 and will prepare some *good first issues*, so you can quickly jump in.**

MT ComparEval Future Steps

The next few days we will deploy a (read-only) demo system. Feel free to check it out when it appears online.

MT-ComparEval

de-nl

	clmtmagic	google	deepl	Add Engine
ted-talks-fbk	translation by clmtmagic	translation by google	translation by deepl	
OPUS-OpenSubtitles2018_90p	translation by clmtmagic	translation by google	Add Task	
OPUS-ECB	translation by clmtmagic	translation by google	Add Task	
DGT-TM	translation by clmtmagic	translation by google	Add Task	
DCEP	translation by clmtmagic	translation by google	Add Task	
OPUS-jrc-V3	translation by clmtmagic	Add Task	Add Task	
OPUS-EMEA	translation by clmtmagic	translation by google	Add Task	
Add Test Set				

View graphical comparison

en-nl

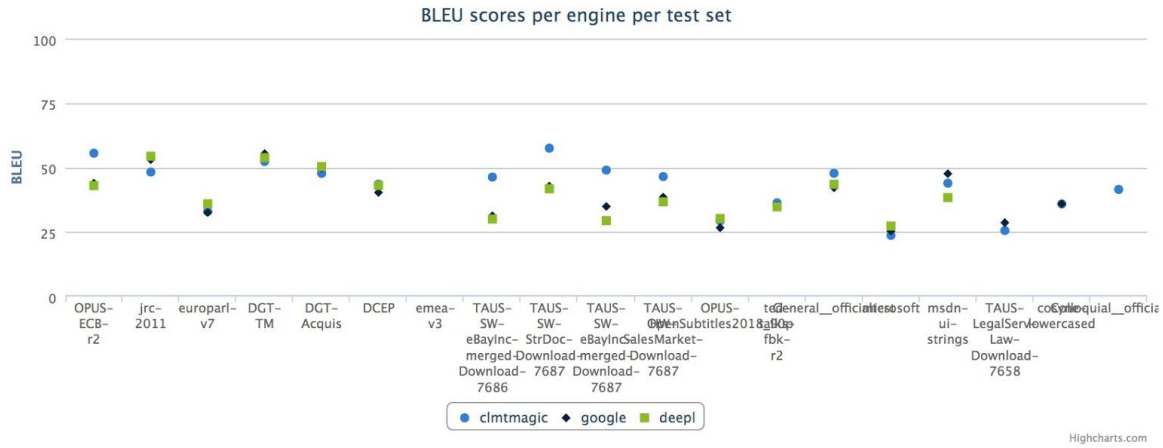
	clmtmagic	google	deepl	Add Engine
Add Test Set				

View graphical comparison

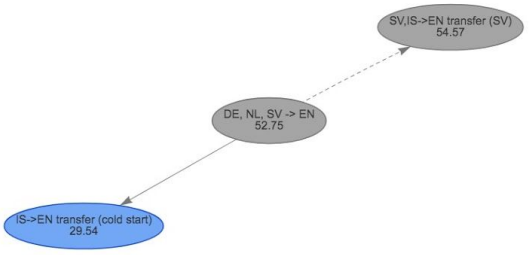
MT-ComparEval

Graphical comparison of engines of language pair en-nl

BLEU scores



Parent-child tree of the IS->EN transfer (cold start) engine



[MT ComparEval Future Steps]

Robustness of NMT

(Tilde)

Recent research has shown that NMT systems are not robust to noise in input data (e.g., Belinkov & Bisk, 2018). Some common types of typical noise in input data are: spelling mistakes, grammar mistakes, noise from speech recognition output (phonetic noise), unknown phenomena. NMT systems are also sensitive to minor changes in input data. E.g., adding a period at the end of a sentence may change the translation of the whole sentence. The idea of this project is to explore methods that can help improving NMT system robustness with respect to some (depending on participants taking part in the project) types of noise in input data.

Belinkov, Y., & Bisk, Y. (2018). Synthetic and Natural Noise Both Break Neural Machine Translation. Proceedings of ICLR 2018.

Terminology Integration in NMT

(Tilde)

The idea of this project is to explore and compare various methods for terminology integration in NMT. Some of the candidate methods to compare: constrained decoding,

factored input data, automatic post-editing, self-adaptive NMT, source-side pre-processing (or terminology injection in the input data), etc.

Unified Training of Machine Translation and Quality Estimation

(Unbabel)

Quality Estimation (QE) models evaluate a machine translation without access to any reference text. Currently this is treated as a task entirely separate from MT, although both MT and QE have at its core a conditional sequence scorer: Conditioned on a source sentence, score target sequences. The key difference is that MT needs to efficiently decode a high scoring sequence, whereas QE has no such requirement.

The effect of this is that QE and MT models have largely similar architectures, differing in the training and inference procedure only (left-to-right decoding for MT, BERT-style bidirectional models for QE)

The goal of this project is to exploit the similarities between the tasks by taking steps towards a unified training of MT and QE models with parameter sharing. The main outcome of this project will be a MT system that simultaneously provides a useful measure of the quality of its translations.

To bootstrap the development, we propose using OpenKiwi[1] (open-source QE with PyTorch) combined with the OpenNMT framework.
As dataset, we propose to focus on the English-German dataset of the WMT19 Quality Estimation shared task, which provides an additional parallel corpus of the same domain consisting of 3 million lines.

Booster Slides: tiny.cc/mi6qbz

[1] Kepler, Fábio et al. "OpenKiwi: An Open Source Framework for Quality Estimation" *Proceedings of the 57th Conference of the Association for Computational Linguistics: System Demonstrations* (2019).

OCELoT: Open, Competitive Evaluation Leaderboard of Translations

(Christian Federmann; chrife@microsoft.com)

Born at MTMA19 in College Park, MD, earlier this year, Project OCELoT aims at implementing an open infrastructure for competitive evaluations, including leaderboards and initially focusing on translations.

Project will start on Tuesday, as I will only arrive in Edinburgh on late Monday.

https://GitHub.com/cfedermann/OCELoT

(Original pitch from MTMA19 below)

Research progress in several machine learning disciplines is tracked via open/competitive leaderboards. While MT has this on a yearly cadence, via the WMT Conference for Machine Translation shared tasks, there is no continuous competition and teams not participating in WMT are left out…

The goal of this project is to build an openly accessible NMT leaderboard where anybody can submit their translation output, both for automatic scoring via SacreBLEU, but also (on some sensible cadence, for the top-n systems to control annotation costs) for human evaluation, adopting WMT's methodology.

You can join this project by 1) contributing code (focusing on Python 3, Django 2, TDD, pylint and Black, amongst others), or by 2) breaking our prototype via submissions of your NMT systems' output. Either will be helpful and much appreciated!

I'll find some human annotation budget for our first batch of participating systems! If this project triggers enough interest from the community, we can look into keeping this alive... Looking forward to your feedback and questions at MTMA19!

Document-level MT

(Unbabel)

Document-level MT is, currently, one of the hot topics in the MT community. Even though SMT systems were already targeting specific phenomena inherent to inter-sentential consistency and agreement, NMT provides a more flexible way of integrating the additional context. In addition, these approaches already showed promising results recently, particularly in WMT19 where some of the approaches (Microsoft and Facebook) where better than a human baseline.

However, these systems are often evaluated in isolation, for instance, the experimental setting of different methods use different language directions (and datasets), preventing a fair comparison between those methods. Therefore, an interesting contribution to the community is assessing the impact of these methods (listed below) under the same experimental settings and evaluating with more than just BLEU. This project requires implementing the methods in a single framework (suggestion fairseq/onmt) and comparing these systems performance in using a number of metrics (listed below) to evaluate specific fine-grained phenomena requiring inter-sentential context.

Next, we summarise the methods, then the metrics (suggestions are welcome), and finally datasets.
Methods:
- Baseline (Transformer)
- Transformer with multi source enc (Voita et. al, 2018)
- Sentence-level transformer with deep integration of APE-like document-level transformer (Voita et. al, 2019)
- Better performance than human baseline at WMT19 (Microsoft, Marcin's approach, or Facebook)
- Hierarchical approach with two levels of attention (Maruf et. al, 2019, sparse attention, or Miculicich et. al, 2018)
- External key-value memory Transformer (adaptation of Tu et. al, 2018)
Metrics:
- Contrastive pronouns (Muller et. al, 2018)
- Selective BLEU (Voita et. al, 2018)
- Lexical cohesion (Miculicich et. al, 2018)
- Suggestions (other metrics: Deixis and Ellipsis -- Voita et. al, 2019; Bawden et. al, 2018, contrastive test-set)
Datasets (EN-DE, at least):
- Subtitles
- Europarl
- TED