mtrain: A Convenience Tool for Machine Translation

Samuel Läubli* and **Mathias Müller*** and **Beat Horat** and **Martin Volk**
{laeubli,mmueller,horat,volk}@cl.uzh.ch
Institute of Computational Linguistics
University of Zurich

Abstract

We present mtrain, a convenience tool for machine translation. It wraps existing machine translation libraries and scripts to ease their use. mtrain is written purely in Python 3, well-documented, and freely available.[1]

Machine translation libraries usually focus on core model training, while data preparation and automatic evaluation are left to the user. This presents a barrier to experimental reproducibility, rapid prototyping, and entry to the field from neighbouring disciplines. In the spirit of the Experimental Management System for Moses (Koehn, 2010), our tool is meant to automate these tasks.

mtrain is designed to handle most aspects of a machine translation experiment: it manages preprocessing, model training, and automatic evaluation. Preprocessing involves automatically splitting a data set into training, validation, and test sets; tokenization; casing; byte-pair encoding; and normalization. On top of these standard preprocessing steps, mtrain can also deal with inline XML markup and intelligently transfer XML tags to translations (Müller, 2017).

Our tool provides training automation for statistical phrase-based models with Moses (Koehn et al., 2007) and neural RNN encoder-decoder models with Nematus (Sennrich et al., 2017). After training, mtrain offers automatic evaluation of translation quality. It outputs the well-known BLEU, TER, and METEOR metrics (Clark et al., 2011). Given a folder that contains trained models, the separate component mtrans can be used to translate from files or standard input.

All steps can be configured with config files or command line options, but default settings already lead to functional baseline systems, making it easier for inexperienced users to use the tool. Going forward, we consider wrapping additional machine translation libraries that are native Python 3, such as Sockeye (Hieber et al., 2017).

[1] https://github.com/ZurichNLP/mtrain

*equal contribution

References

Clark, Jonathan H, Chris Dyer, Alon Lavie, and Noah A Smith. 2011. Better hypothesis testing for statistical machine translation: Controlling for optimizer instability. In *Proceedings of ACL*, pages 176–181.

Hieber, Felix, Tobias Domhan, Michael Denkowski, David Vilar, Artem Sokolov, Ann Clifton, and Matt Post. 2017. Sockeye: a toolkit for neural machine translation. *arXiv preprint*, arXiv:0902.0885.

Koehn, Philipp, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, et al. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of ACL*, pages 177–180.

Koehn, Philipp. 2010. An experimental management system. *The Prague Bulletin of Mathematical Linguistics*, 94:87–96.

Müller, Mathias. 2017. Treatment of markup in statistical machine translation. In *Proceedings of the Third Workshop on Discourse in Machine Translation*, pages 36–46.

Sennrich, Rico, Orhan Firat, Kyunghyun Cho, Alexandra Birch, Barry Haddow, Julian Hitschler, Marcin Junczys-Dowmunt, Samuel Läubli, Antonio Valerio Miceli Barone, Jozef Mokry, and Maria Nadejde. 2017. Nematus: a toolkit for neural machine translation. In *Proceedings of EACL*, pages 65–68.

Pérez-Ortiz, Sánchez-Martínez, Esplà-Gomis, Popović, Rico, Martins, Van den Bogaert, Forcada (eds.)
Proceedings of the 21st Annual Conference of the European Association for Machine Translation, p. 357
Alacant, Spain, May 2018.