Project 2: Neural Machine Translation

April 29, 2018

This project will help you learn and implement a neural machine translation model with attention. In summary, your task is to:

- Pre-process the training, validation, and test data;
- Implement the model as described below;
- Optionally implement one or more of the `extra`'s;
- Evaluate your model on the test data;
- Write a report on the entire process.

1 Neural Machine Translation

Implement the following, using English–French parallel data (translations from French into English):

1. Preprocessing:
   a) Tokenisation;
   b) Lowercasing or truecasing;
   c) Byte-pair encodings (BPE);
2. Seq2seq with positional embeddings (without an RNN encoder) (e.g. attention is all you need paper);
   a) ***Extra***. Use a different encoder (GRU or LSTM);
3. Attention:
   a) Dot product;
   b) ***Extra***. Bilinear;

4. Regularization with dropout;
5. Evaluation (BLEU, Meteor, TER);
6. ***Extra***. Attention visualization;
7. ***Extra***. Beam search decoder;
8. ***Tips***: https://github.com/neubig/nmt-tips.

2 Data

All relevant data (including details about file formats) are available from https://uva-slpl.github.io/nlp2/projects.html.

In this project, you will work with a parallel corpus based on the Flickr30k data set.[1] This corpus is called Multi30k (more information here[2]) and the English-French parallel sentences used for training, validation and testing your models are publicly available.[3]

We are making available *training* data (which you can use to perform parameter estimation), *validation* data (which you can use to debug your implementation as well as to perform model selection), and finally in due time *test* data (which you will use to conduct your final empirical comparison).

3 Report

You should use LaTeX for your report, and you should use the ACL template available from http://acl2017.org/downloads/acl17-latex.zip (unlike the template suggests, your submission should not be anonymous).

We expect short reports (2–4 pages plus references) written in English. The typical submission is organised as follows:

- **Abstract**: conveys scope and contributions;
- **Introduction**: present the problem and relevant background;
- **Model**: technical description of models;
- **Experiments**: details about the data, experimental setup and findings;
- **Conclusion**: a critical take on contributions and limitations.

[1] For more information, see http://web.engr.illinois.edu/~bplumme2/Flickr30kEntities/.
[2] https://github.com/multi30k/dataset
[3] https://github.com/uva-slpl/nlp2/tree/gh-pages/resources/project_nmt/data.

4 Submission

You should submit a `.tgz` file containing a folder (folder name `lastname1.lastname2`) with the report as a single `pdf` file (filename: `report.pdf`). Your report may contain a link to an open-source repository (such as github), but please do not attach code or additional data to your submission. You can complete your project submission on Blackboard.

5 Assessment

Your report will be assessed by two independent reviewers according to the following evaluation criteria:

1. **Scope** (max 2 points): Is the problem well presented? Do students understand the challenges/contributions?
2. **Theoretical description** (max 3 points): Are the models presented clearly and correctly?
3. **Empirical evaluation** (max 3 points): Is the experimental setup sound/convincing? Are experimental findings presented in an organised and effective manner?
4. **Writing style** (max 2 points): use of LaTeX, structure of report, use of tables/figures/plots, command of English.
5. **Extra**: surprise...

6 Resources

A non-exaustive list of resources that might help you along the way:

1. **Moses scripts**:
   (a) Tokenizer: https://github.com/moses-smt/mosesdecoder/blob/master/scripts/tokenizer/tokenizer.perl;
   (b) Lowercaser: https://github.com/moses-smt/mosesdecoder/blob/master/scripts/tokenizer/lowercase.perl;
   (c) Truecasing tools: https://github.com/moses-smt/mosesdecoder/tree/master/scripts/recaser;
   (d) BLEU: https://github.com/moses-smt/mosesdecoder/blob/master/scripts/generic/multi-bleu.perl;
   (e) METEOR: http://www.cs.cmu.edu/~alavie/METEOR/;

(f) TER: https://github.com/jhclark/tercom;

2. **Byte-pair encoding** scripts: https://github.com/rsennrich/subword-nmt.