Natural Language Understanding with Distributed Representations

Assignment 4

Felipe Ducau
fnd212@nyu.edu

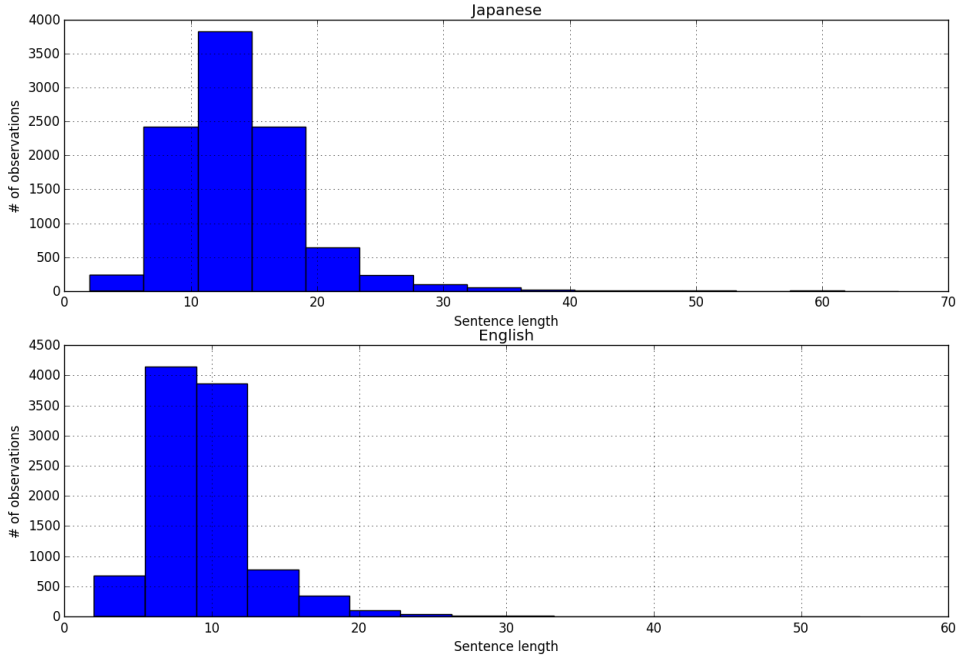
1 Data Preparation

The data [1] is composed of 11000 sentences in japanese with their corresponding translation in english. The training set contains 10000 sentences while both validation and test sets contain 500 sentences each.

Since the data is already in lowercase and the punctuation symbols already appear separated from other words by blank spaces, the only preprocessing used for the experiments of this assignment was to tokenize the words splitting them by spaces.

The size of the vocabulary in the english train set is 102207 words and in the japanese train set is 239126. From these words, to train our model we only keep the top 40000 words for each language.

The maximum longitude of sentence used along the experiments is 50, trimming those which were larger. The maximum sentence length along the train set is 66 for japanese sentences and 54 for english sentences The figure below shows the distribution of sentence lengths for each vocabulary.



2 Neural Machine Translation

The following parameters are common to both models implemented (Simple encoder-decoder and Attention-based NMT):

- Word embedding dimension: 128.
- GRU context vector dimension: 256.
- Early stopping was impelemented with 10 of patience in the validation perplexity score.

[1] https://github.com/neubig/nmt-tips

- Maximum longitude of a sentence: 50.
- Encoder RNN unit: GRU / Depth 1.
- Decoder RNN unit: GRU / Depth 1.
- Batch size: 100.
- Optimization method: Adadelta [2].
- Initial learning: 0.01, then adaptive acording to Adadelta.
- Translation generation procedure: Greedy.

All the code for this assignment was implemented in Theano and it is based on parts of the code we are developing for our final project which is itself built upon `https://github.com/laulysta/nmt`.

2.1 Simple Encoder-Decoder Model

The model used for this section was a basic Encoder-Decoder architecture with GRU units according to Cho et al 2014 [1]

2.1 Simple Encoder-Decoder Model

For this section, instead of using the output of the encoder as the input to the decoder, we implemented an attention mechanism in which the encoder looks at all the intermediate outputs of the encoder and create a weighted average of them according to its current input and memory state.

The implementation follows the one in `https://github.com/nyu-dl/dl4mt-tutorial` in which we use a bidirectional (forward and backward) encoder with a conditional GRU with attention mechanism for the decoder.

3 Results

Table 1 summarizes the results for the two different implementations.

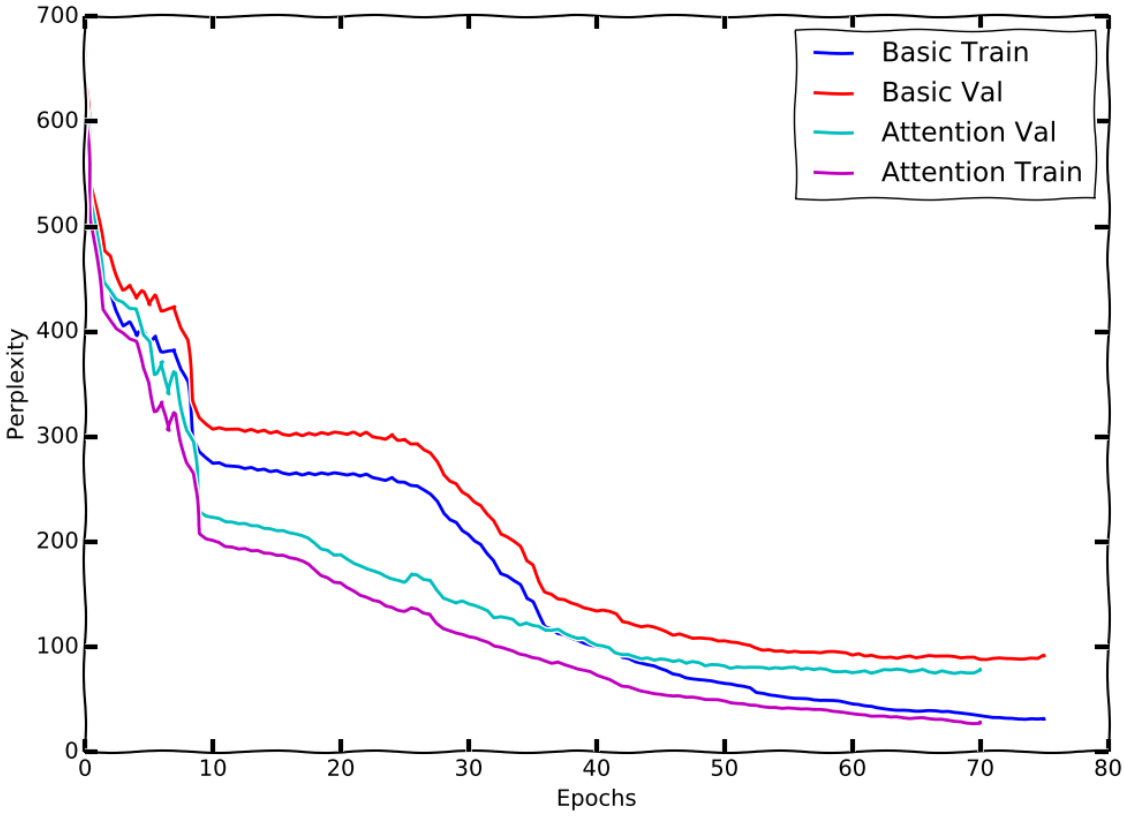
Model	Train Perplexity	Validation Perplexity	Test BLEU
Encoder-Decoder	34.66	88.52	0.14
Attention-based NMT	30.03	75.19	0.18

Table 1: Results.

The BLEU score in the table corresponds to the corpus level BLEU score [2] which is smaller than the average of the BLEU score of the sentences. As expected, the basic model performs worth in terms of BLEU. The small value of the score is most probably consequence of the small size of the dataset.

The table below shows the training curves for both models. Even though the performance for the attention model is better overall, we note that the difference between the performance en train for both models decrease with the number of iterations.

[2] `http://www.nltk.org/_modules/nltk/translate/bleu_score.html`



References

[1] Kyunghyun Cho et al. "Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation". In: *CoRR* abs/1406.1078 (2014). URL: http://arxiv.org/abs/1406.1078.

[2] Matthew D. Zeiler. "ADADELTA: An Adaptive Learning Rate Method". In: *CoRR* abs/1212.5701 (2012). URL: http://arxiv.org/abs/1212.5701.