Train Better Models Faster

Curriculum Learning and Intelligent Hyperparameter Search for Neural Machine Translation

Xuan Zhang

Advised by Kevin Duh

Dec 07 2018
Neural Machine Translation (NMT) models are data-hungry monsters and expensive to train.

Can we train better NMT models faster?

I. Curriculum Learning — Improve sample efficiency (co-advised by Marine Carpuat, University of Maryland)

- In-Domain Training
- Domain adaptation

II. Intelligent Hyperparameter Search — Speed up model selection

- Auto-Tuning
- Representative Subcorpus
Curriculum Learning

In Machine Learning:
• Introduce gradually more difficult examples to the learner.
• Perceptron, SGD and CNN can converge faster.

Can Seq2Seq NMT models also benefit from it?
Curriculum Learning

In Machine Learning:
• Introduce gradually more difficult examples to the learner.
• Perceptron, SGD and CNN can converge faster.

Can Seq2Seq NMT models also benefit from it?
Curriculum Learning for Neural Machine Translation

What is an easy-to-learn example in NMT?

I. Linguistic features
   - Sentence Length
   - Word Frequency Rank (max, average)

II. Transfer knowledge from a teacher model
   - One-best Score
Curriculum Learning for Neural Machine Translation

What is the curriculum training strategy?

- **Probabilistic curriculum training strategy** (our approach)

  - Sentence ranking
  - Data sharding
Curriculum Learning for Neural Machine Translation

What is the curriculum training strategy?

- **Probabilistic curriculum training strategy** (our approach)
  - Sentence ranking
  - Data sharding
What is the curriculum training strategy?

- **Probabilistic curriculum training strategy** (our approach)

- Sentence ranking
- Data sharding
Curriculum Learning for Neural Machine Translation

What is the curriculum training strategy?

- Probabilistic curriculum training strategy (our approach)

- Sentence ranking
- Data sharding
Curriculum Learning for Neural Machine Translation

What is the curriculum training strategy?

- **Probabilistic curriculum training strategy** (our approach)

  - Sentence ranking
  - Data sharding
Curriculum Learning for Neural Machine Translation

What is the curriculum training strategy?

- **Probabilistic curriculum training strategy** (our approach)
  - Sentence ranking
  - Data sharding
Curriculum Learning for Neural Machine Translation

What is the curriculum training strategy?

- **Probabilistic curriculum training strategy** (our approach)
  - Sentence ranking
  - Data sharding
Curriculum Learning for Neural Machine Translation

What is the curriculum training strategy?

- **Probabilistic curriculum training strategy** (our approach)

  - Sentence ranking
  - Data sharding

shards

| easy | difficult |

NMT model
Curriculum Learning for Neural Machine Translation

What is the curriculum training strategy?

- **Probabilistic curriculum training strategy** (our approach)

  - Sentence ranking
  - Data sharding
What is the curriculum training strategy?

- **Probabilistic curriculum training strategy** (our approach)

  - Sentence ranking
  - Data sharding

converged!
Curriculum Learning for Neural Machine Translation

What is the curriculum training strategy?

- Probabilistic curriculum training strategy (our approach)

- Sentence ranking
- Data sharding
- Training on a subset of shards in a phase
- Including more difficult shards gradually
- Presenting order is not deterministic:
  1. shard shuffling within a phase
  2. bucketing, mini-batching within a shard

... until converged
Curriculum Learning for Neural Machine Translation

- Performance of curriculum learning strategies with different difficulty criteria

<table>
<thead>
<tr>
<th>Training Time (1000 batches)</th>
<th>baseline</th>
<th>max wd freq(de)</th>
<th>max wd freq(en)</th>
<th>max wd freq (deen)</th>
<th>ave wd freq(de)</th>
<th>ave wd freq(en)</th>
<th>ave wd freq (deen)</th>
<th>sentence len(de)</th>
<th>sentence len(en)</th>
<th>sentence len (deen)</th>
<th>one-best score</th>
</tr>
</thead>
<tbody>
<tr>
<td>BLEU</td>
<td>28.1</td>
<td>25.2</td>
<td>27.6</td>
<td><strong>28.1</strong></td>
<td>28.2</td>
<td>27.8</td>
<td>27.3</td>
<td>26.6</td>
<td>27.6</td>
<td>27.0</td>
<td>27.0</td>
</tr>
<tr>
<td>Training Time (1000 batches)</td>
<td>73</td>
<td>57</td>
<td>63</td>
<td><strong>56</strong></td>
<td>72</td>
<td>84</td>
<td>62</td>
<td>78</td>
<td>151</td>
<td>113</td>
<td>56</td>
</tr>
</tbody>
</table>
Curriculum Learning for Domain Adaptation

- Score and rank training examples by their similarity to in-domain data
- Same curriculum training strategy can be applied
- Similarity: data selection methods
  (Moore-Lewis score, cynical data selection)
Curriculum Learning for Domain Adaptation

• Data Selection for Domain Adaptation

1. Score non-domain-specific sentences based on their similarity to in-domain data
2. Sort the sentences
3. Select training data from the non-domain-specific data using a cut-off threshold on the resulting scores
4. Build domain-specific NMT system
Curriculum Learning for Domain Adaptation

• Domain similarity scoring

Moore-Lewis Score (cross-entropy difference score)

\[ H_I(s) - H_N(s) \]

- \( H \): cross-entropy
- \( I \): In-domain
- \( N \): Non-domain-specific
- \( s \): a non-domain-specific sentence

A lower ML score indicates \( s \) is more like the in-domain data and less like the non-domain-specific data
Curriculum Learning for Domain Adaptation

• Domain similarity scoring

Cynical Data Selection (Incremental greedy selection)

\[ H_n = - \sum_{v \in V_I} \frac{C_I(v)}{W_I} \log \frac{C_n(v)}{W_n} \quad P(n) \log Q(n) \]

\[ H_{n+1} = H_n + \Delta H \]

\[ \Delta H_{n \rightarrow n+1} = \log \frac{W_n + w_{n+1}}{W_n} \]

\[ = \sum_{w \in V_I} \frac{C_I(v)}{W_I} \log \frac{C_n(v)}{C_n(v) + c_{n+1}(v)} \]

* \( W_n \) is the total number of word tokens in the previous selected lines

* \( C_n(v) \) is the count of word \( v \) in the previous selected lines
Curriculum Learning for Domain Adaptation

- Evaluation on Continued Training Setup

- General Domain Data
- Domain Specific Model
- Initialization
- In-Domain Data
- Standard training (baseline)
- Continuation Training
- Non-Domain-Specific Data
- Curriculum Learning
- Similarity Scoring

- Generic Model
Curriculum Learning for Domain Adaptation

- **Evaluation on Continued Training Setup**

  Improves BLEU by **5%~10.4%** (up to **3.22** BLEU points).
Curriculum Learning for Domain Adaptation

Where does the gain come from?

**S4 Error Analysis** (word level translation error)

- \( f_i \): Source word
- \( e_j \): Reference translation of \( f_i \)
- \( H_i \): Output translation of \( f_i \)

**ERROR**

\[ e_j \not\in H_i \]

**CORRECT**

\[ e_j \in H_i \]

1. **SEEN**

\[ f_i \not\in \text{training corpus} \]

2. **SENSE**

\[ f_i \in \text{training corpus}, \ e_j \not\in \text{training corpus} \]

3. **SCORE**

\[ (f_i, e_j) \in \text{training corpus} \]

4. **SEARCH** a translation error due to pruning (a small beam size)
Curriculum Learning for Domain Adaptation

Where does the gain come from?

S4 Error Analysis (word level translation error)
Curriculum Learning for Domain Adaptation

Support our hypothesis that it is beneficial to train on examples that are closest to in-domain first and to use a probabilistic curriculum.
Can we train better NMT models faster?

I. Curriculum Learning — Improve sample efficiency

- In-Domain Training
  - Can improve sample efficiency at early stage of training
  - No clear pattern found
- Domain Adaptation
  - Consistently outperform the standard continued training model
  - Improve SCORE and SENSE errors

II. Intelligent Hyperparameter Search — Speed up model selection

- Auto-Tuning
- Representative Subcorpus
Auto-tuning

- Exhaustive hyperparameter search is time-consuming
- Automatic system tuning process using CMA-ES
Representative Subcorpus

Can we find a small representative subset of the large training corpus, so that the hyperparameter tuned on the representative subset can generalize to the original large dataset?

- Let’s first try **uniform sampling** from the large corpus.
Representative Subcorpus

Can we find a small representative subset of the large training corpus, so that the hyperparameter tuned on the representative subset can generalize to the original large dataset?

- Let’s first try **uniform sampling** from the large corpus.
  - The hyperparameters tuned on uniform sampled subcorpus can generalize to the large corpus
  - Average time saved: **60 clock hours**

(1/2 of the training time spent by models trained on large corpus: >120 hours)
Representative Subcorpus

Can this be further improved by selecting the subcorpus in a more clever way?

- Sampling by n-gram distribution
  - Representative subcorpus:
    sentences containing only the most frequent words (top 1/256);
    1/2 of the original corpus size
  - Performance ranking holds on both large and small corpus
  - Average time saved: around **100 clock hours**
    (3/4 of the training time spent by models trained on large corpus: >120 hours)
  - Faster than uniform sampling
Can we train better NMT models faster?

I. Curriculum Learning — Improve sample efficiency

- In-domain training
  - Can improve sample efficiency at early stage of training
  - No clear pattern found
- Domain adaptation
  - Consistently outperform the standard continued training model
  - Improve SCORE and SENSE error

II. Intelligent Hyperparameter Search — Speed up model selection

- Auto-tuning
- Representative subcorpus
• Curriculum learning and auto-tuning implementations are public available at:

https://github.com/kevinduh/sockeye-recipes