Efficient Machine Translation

Lei Li
University of California Santa Barbara
2021/10/10
Cross Language Barrier with Machine Translation

- Foreign Media
- Global Conferences
- Tourism
- International Trade

The latest version will launch in just a few months.
Neural Machine Translation

• Transformer as commonly used backbone architecture for MT.
• 50 - 100m parameters
• Huge computation: 670 GPU hours for training [Vaswani et al 2017].
Training NMT gets more expensive!

- Attention GPU is all you need

<table>
<thead>
<tr>
<th>model</th>
<th>Size (M)</th>
<th>Total Time (GPU hr)</th>
<th>Train Once (GPU hr)</th>
<th>Infer (ms)</th>
<th>Carbon Footprint (car year)</th>
</tr>
</thead>
<tbody>
<tr>
<td>mRASP (EMNLP20)</td>
<td>60</td>
<td>38k</td>
<td>384</td>
<td></td>
<td></td>
</tr>
<tr>
<td>mRASP2 (ACL 21)</td>
<td>450</td>
<td>128k</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LaSS (ACL 21)</td>
<td>60</td>
<td>41k</td>
<td>384</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LUT (AAAI 21)</td>
<td>144</td>
<td>22k</td>
<td>72</td>
<td>150</td>
<td></td>
</tr>
<tr>
<td>COSTT (AAAI 21)</td>
<td>55</td>
<td>22k</td>
<td>72</td>
<td>140</td>
<td></td>
</tr>
<tr>
<td>Chimera (ACL 21)</td>
<td>165</td>
<td>59k</td>
<td>320</td>
<td>160</td>
<td></td>
</tr>
<tr>
<td>XSTNet (Interspeech21)</td>
<td>152</td>
<td>24k</td>
<td>240</td>
<td>140</td>
<td></td>
</tr>
</tbody>
</table>
Affordable and Green MT

• Training NMT models are computationally expensive.
• How to speed up MT training, and inference?
• How to reduce energy consumptions during MT training?
1. Algorithm: Learning Compact Vocabulary for NMT
   – Small vocabulary with improved performance at 100x faster!

2. Model: Parallel Generation
   – Translate at equal or better quality with 10x speedup!

3. Computing: Hardware Acceleration for training and inference
   – Faster than Tensorflow & Pytorch at 14x speedup!
Vocabulary Learning via Optimal Transport for Neural Machine Translation

joint w/ Jingjing Xu¹  Hao Zhou¹  Chun Gan¹  Zaixiang Zheng¹
Vocabulary is Fundamental and Important

NER  Sentiment Analysis  Translation  Dialog  Summarization

Output

Encoder

Tokenize

Input Sentence

Vocab

Generate w/ Vocab

Decoder

Token ID

<table>
<thead>
<tr>
<th>Token</th>
<th>ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>0</td>
</tr>
<tr>
<td>es</td>
<td>1</td>
</tr>
<tr>
<td>cat</td>
<td>2</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
Which Vocabulary is Better?

**Word level**

The most eager is Oregon which is enlisting 5,000 drivers in the country

**Char level**

Th e m o s t e a g e r i s O r e g ...  

**Sub-word level**

Th e m o s t e a g e r i s O r e g on which is en listing 5,000 drivers in the country

Sub-word vocabulary is the dominant choice

* With normal-size data
Why is Sub-word (BPE) superior? Theoretically

- Information theory:
  - Compress the message into compact representation
  - Fewest bits to represent both sentence and vocabulary
  - Char-level vocab ==> text sequence will be long
  - Word-level vocab ==> vocab will be large and still OOV

- Entropy:
  - How much information in each token

- Intuition:
  - Reduced entropy (bits-per-char) ==> Better Vocab
  - Even better vocab?
Information-theoretic Vocabulary Evaluation

• Normalized Entropy
  – Information-per-char (IPC)
    
    \[ H(v) = -\frac{1}{l_v} \sum_{i \in v} P(i)\log P(i) \]

  – It represents Semantic-information-per-char
    
    • Smaller IPC is better. Easy to differentiate (therefore easy to generate)

<table>
<thead>
<tr>
<th>Token</th>
<th>count</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>200</td>
</tr>
<tr>
<td>e</td>
<td>90</td>
</tr>
<tr>
<td>c</td>
<td>30</td>
</tr>
<tr>
<td>t</td>
<td>30</td>
</tr>
<tr>
<td>s</td>
<td>90</td>
</tr>
</tbody>
</table>

\[ H(v) = 1.37 \]

VS

<table>
<thead>
<tr>
<th>Token</th>
<th>count</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>100</td>
</tr>
<tr>
<td>aes</td>
<td>90</td>
</tr>
<tr>
<td>cat</td>
<td>30</td>
</tr>
</tbody>
</table>

\[ H(v) = 0.14 😊 \]
Which vocabulary is better?

Sub-word level vocabulary with 1K tokens (BPE-1K)

Sub-word level vocabulary with 10K tokens (BPE-10K)

Sub-word level vocabulary with 30K tokens (BPE-30K)

From the perspective of size, BPE-1K seems to be better but longer sequence

* With normal-size data
Which Vocabulary is Better?

Sub-word level vocabulary with 1K tokens (BPE-1K)

Sub-word level vocabulary with 10K tokens (BPE-10K)

Sub-word level vocabulary with 30K tokens (BPE-30K)

From the perspective of entropy, BPE-30K seems to be better

* With normal-size data
Evaluating Vocabulary Quality is Expensive

Full training and testing are required to find the optimal vocabulary!
Trading IPC with Size

• Value: **IPC**

• Cost: **size**

• Marginal utility of information for Vocabulary (MUV)
  – Negative **gradients** of IPC to size
  – How many value does each unit-of-cost bring?
MUV is good indicator for MT performance

- Cost-effective point in MUV curve (maximum MUV)
  - ==> best BLEU

MUV Indicates MT Performance

- MUV and BLEU are **correlated** on two-thirds of tasks
- A good coarse-grained evaluation metric!

---

Maximizing Marginal Utility of Vocab

• Goal: finding the optimal vocabulary

  Finding the optimal vocabulary  \[\rightleftharpoons\]  Maximizing MUV

• Naive solution:
  – Exhaustive Search for vocabulary with max MUI

• How to search over a huge discrete space?

Problem Reduction

• Best BLEU ==> Max MUV ==> Optimal Transport

Min cost to Transport soldiers from bases to frontlines
Optimal Transport

Easy solution: split the task with proportions 120:90:90 = 4:3:3
Optimal Transport

Easy solution: split the task with proportions
120:90:90 = 4:3:3
Vocabulary building as Transportation

• Adding one new token means:
  – Transport character frequency to token frequency

  – a b c d e a b d c e f: 2 a 2 b

  – ab c d e ab d c e f: 0 a 0b and 2 ab
VOLT Formulation

Transport chars to tokens

- a (60)
- c (60)
- cat (60)
- at (90)
- ca (90)
- t (150)
Not all tokens can get chars
Not all tokens can get chars
VOLT Formulation

Not all tokens can get chars

Each Transportation Defines a Vocabulary

Reducing MUV Optimization to OT

• The vocabulary with the maximum MUV
  – Maximum gap between IPC of a vocabulary (with size t) and that of a smaller vocabulary (with size <t)
  – \( \max \left( \max H(V_{t+1}) - \max H(V_t) \right) \)

• Intractable, instead to maximize upper-bound of gap \( (H(V_{t+1}) - H(V_t)) \)

• \( \Rightarrow \) \( \max(\max H(V_{t+1}) - \max H(V_t)) \)

• Finding \( \max H(V_t) \) \( \Rightarrow \) Optimal Transport

Find the transportation matrix (\(=\text{vocab}\)) with lowest cost (\(-\text{MUV}\))

\[
\begin{align*}
\forall j \in \{a, b, c\}, \sum_{i \in \{ab, bc, a\}} p_{i,j} &= 1 \\
\forall i \in \{ab, bc, a\}, \sum_{j \in \{a, b, c\}} p_{i,j} - p_i &\leq \varepsilon
\end{align*}
\]

**Cost Function**

\[
C(P) = -H(P) + \sum_{j \in \{a, b, c\}, \ i \in \{ab, bc, a\}} p_{i,j}D_{i,j}
\]

**Transportation matrix** \(P\)

<table>
<thead>
<tr>
<th></th>
<th>cat</th>
<th>at</th>
<th>tea</th>
</tr>
</thead>
<tbody>
<tr>
<td>cat</td>
<td>20</td>
<td>10</td>
<td>0</td>
</tr>
<tr>
<td>at</td>
<td>20</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>tea</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>20</td>
<td>10</td>
<td>0</td>
</tr>
</tbody>
</table>

**Cost matrix** \(D\)

<table>
<thead>
<tr>
<th></th>
<th>cat</th>
<th>at</th>
<th>tea</th>
</tr>
</thead>
<tbody>
<tr>
<td>cat</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>at</td>
<td>1</td>
<td>(\infty)</td>
<td>(\infty)</td>
</tr>
<tr>
<td>tea</td>
<td>(\infty)</td>
<td>(\infty)</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>1</td>
<td>(\infty)</td>
</tr>
</tbody>
</table>

**Sinkhorn Algorithm** [Gabriel Peyré et al.]

VOLT finds better vocabulary on Bilingual MT

VOLT finds better vocabulary on Bilingual MT

VOLT finds better vocabulary on Bilingual MT

Transformer architecture

WMT De-En
- BLEU (+)
- Size (K) (-)
- BPE-30K (Widely-adopted)
- VOLT

TED Es-En
- BLEU (+)
- Size (K) (-)
- BPE-30K (Widely-adopted)
- VOLT

TED PTbr-En
- BLEU (+)
- Size (K) (-)
- BPE-30K (Widely-adopted)
- VOLT

TED Fr-En
- BLEU (+)
- Size (K) (-)
- BPE-30K (Widely-adopted)
- VOLT

TED Ru-En
- BLEU (+)
- Size (K) (-)
- BPE-30K (Widely-adopted)
- VOLT

TED He-En
- BLEU (+)
- Size (K) (-)
- BPE-30K (Widely-adopted)
- VOLT

TED Ar-En
- BLEU (+)
- Size (K) (-)
- BPE-30K (Widely-adopted)
- VOLT

TED It-En
- BLEU (+)
- Size (K) (-)
- BPE-30K (Widely-adopted)
- VOLT

VOLT Finds Better Vocabulary on Multilingual MT

Transformer architecture

BLEU

VOLT Generalizes Well to Other Architectures

**VOLT: A Green Vocabulary Learning Solution**

<table>
<thead>
<tr>
<th>Carbon Emission</th>
</tr>
</thead>
<tbody>
<tr>
<td>BPE-Search</td>
</tr>
<tr>
<td>VOLT-search</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>BLEU</th>
</tr>
</thead>
<tbody>
<tr>
<td>BPE-Search</td>
</tr>
<tr>
<td>VOLT</td>
</tr>
</tbody>
</table>

VOLT: A Green Vocabulary Learning Solution

Carbon Emission

<table>
<thead>
<tr>
<th>Method</th>
<th>Carbon Emission</th>
</tr>
</thead>
<tbody>
<tr>
<td>BPE-Search</td>
<td>384 GPU hours</td>
</tr>
<tr>
<td>VOLT-search-eval</td>
<td>0.5 CPU hours + 30 GPU hours</td>
</tr>
</tbody>
</table>

BLEU

<table>
<thead>
<tr>
<th>Method</th>
<th>BLEU Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>BPE-Search</td>
<td>29.9</td>
</tr>
<tr>
<td>VOLT</td>
<td>29.8</td>
</tr>
</tbody>
</table>

How to reduce this?
Glancing Transformer for Non-autoregressive Neural Machine Translation

Joint w/ Lihua Qian, Hao Zhou, Yu Bao, Mingxuan Wang, Lin Qiu, Weinan Zhang, Yong Yu
Transformer is Autoregressive

- Autoregressive models generate sentences sequentially
  
  \[ p(Y|X; \theta) = \prod_{t=1}^{T} p(y_t|y_{<t}, X; \theta) \]

- The conditional probability is factorized successively

- Human-style translation is slow. Machine does not have to mimic human!
Wild idea: Parallel Generation?

• Non-autoregressive models generate all the tokens in parallel

很多树 ➔ a lot of trees

• Conditional independence assumption

\[ p(Y|X; \theta) = \prod_{t=1}^{T} p(y_t|X; \theta) \]
Model architecture

Autoregressive decoder  Non-autoregressive decoder

Gu et al, NAT, ICLR 2018
Why Non-autoregressive?

1. Faster decoding in non-autoregressive translation (NAT)

   a lot of trees

2. Capturing bidirectional context for generation

   a lot of trees
Challenge: Inferior Quality of NAT

- One input -> multiple target
  - a lot of trees
  - a great many trees

- Inconsistency problem in parallel generation
  - great many
  - lot of
  - a great of trees

---

**BLEU**

<table>
<thead>
<tr>
<th></th>
<th>WMT14 EN-DE</th>
<th>WMT14 DE-EN</th>
</tr>
</thead>
<tbody>
<tr>
<td>NAT-base</td>
<td>18</td>
<td></td>
</tr>
<tr>
<td>Transformer</td>
<td>33</td>
<td>27</td>
</tr>
</tbody>
</table>

**WMT14 EN-DE**

**WMT14 DE-EN**
Key Intuition: Word interdependency

- Learning *word interdependency* in the *target sentence* is crucial for generating fluent sentences.
- Non-autoregressive models lack an effective way of dependency learning.
Autoregressive models

- predict the next tokens conditioned on the input target tokens (left-to-right)
Learning Word Interdependency

Iterative-NAT

• predict the randomly masked tokens based on unmasked tokens

rely on multiple decoding iterations, therefore does not gain speedup!

Lee et al. Deterministic Non-Autoregressive Neural Sequence Modeling by Iterative Refinement. EMNLP 2018.
New Idea for Dependency learning

\[ L_\theta = - \log p(Y|X; \theta) \]

- Glancing Language Model (GLM)
  - A gradual training method
  - Learning word interdependency for **single-pass** parallel generation

Glancing Language Model

- Perform two decoding during training
  1. Glancing Sampling (the first decoding):
     - Based on the prediction, replace part of the decoder inputs with sampled target words
  2. Optimization (the second decoding):
     - Learn to predict the remaining words with the replaced decoder inputs

- Based on the prediction, replace part of the decoder inputs with sampled target words

- Learn to predict the remaining words with the replaced decoder inputs

During training, the sampling number of target words decreases gradually.

GLAT boosts Translation Quality significantly! + 5 BLEU!

GLAT approaches Transformer quality!

- GLAT achieves high quality translation while keeping high inference speed-up (8x~15x)

GLAT in Real Competition

GLAT achieve the Top score in WMT21 En-De and De-En! The first NAT system to do so!

GLAT is the first production NAT system!

- Already deployed online in VolcTrans and serving English-Japanese

Tiktok caption translation
LightSeq: A High Performance Library for Transformers

Joint w/ Xiaohui Wang, Ying Xiong, Yang Wei, Xian Qian, Mingxuan Wang and community contributors
Need for Hardware Acceleration

• What about Transformer computing?
  – Transformers are still widely used in many sequence processing and generation tasks.

• Large number of parameters cause the high latency in training and inference.

• Current computation libraries are insufficient.
LightSeq: A high-performance library

- **Efficient**
  - LightSeq achieves up to 14x speedup compared with TensorFlow and PyTorch.

- **Functional**
  - LightSeq supports more architecture variants and different search algorithms.

- **Convenient**
  - LightSeq is easy to use without any code modification.
  - Seamless porting from Tensorflow, Pytorch, Hugginface, Fairseq
• LightSeq greatly reduces the proportion of kernels other than GEMM.

• LightSeq outperforms others in most cases, especially in large batch size.
Faster Text Generation w/ LightSeq

- LightSeq outperforms others in most cases

Summary for Efficient MT

- **Algorithm: VOLT**
  - Learning Compact Vocabulary for NMT
  - Small vocabulary with improved performance at 100x faster!
  - Green solution: 30mins on only one cpu.

- **Model: GLAT**
  - Parallel Generation really works for the first time!
  - Translate at equal or better quality with 10x speedup!
  - Deployed in production

- **Computing: LightSeq**
  - Hardware Acceleration for training and inference
  - 14x faster than Tensorflow & Pytorch!
Towards Green MT

• Many challenges remaining!
• Propose new metric: Best value MT
  – GFlops or carbon footprint for model development
• Hardware acceleration for GLAT and other NAT?
• Low-end hardware?
• Taming the model size?
Thanks!

• Code:
  – VOLT: [https://github.com/Jingjing-NLP/VOLT](https://github.com/Jingjing-NLP/VOLT)
  – GLAT: [https://github.com/FLC777/GLAT](https://github.com/FLC777/GLAT)

• Open Source Library

Contact: lilei@cs.ucsb.edu