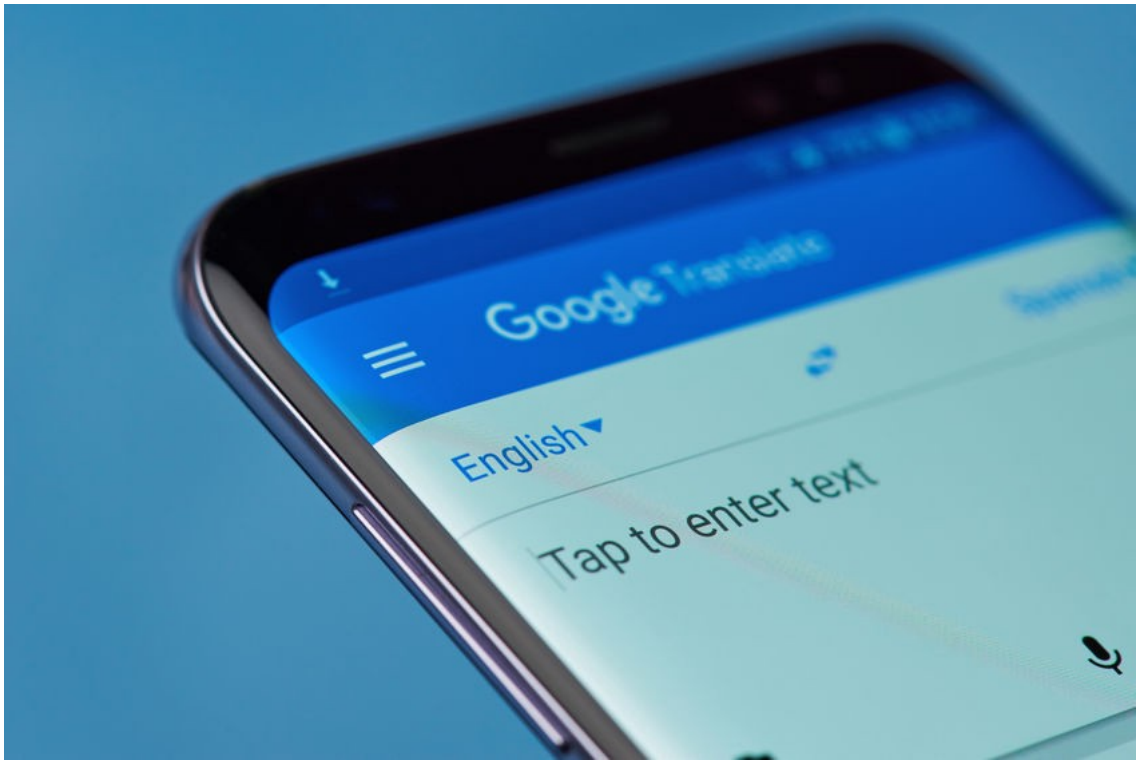
Sik-Ho Tsang · Follow
Nov 20, 2021 · 8 min read · Listen

Review — Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation (GNMT)

GNMT, Wordpiece Model, Using Deep LSTM With Residual Connections



In this story, **Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation**, (GNMT), by Google, is reviewed. In this paper:

- **A deep LSTM network** with 8 encoder and 8 decoder layers using residual connections as well as attention connections from the decoder network is used.
- **Parallelism** is used to reduce the training time.
- **Low-precision arithmetic** is used for faster inference.
- **Wordpiece-based translation** is used instead of word or character-based.
- **Beam search** technique employs a **length-normalization** procedure and uses a **coverage penalty**.
- **Reinforcement learning** is used to refine the model to directly optimize the BLEU score.

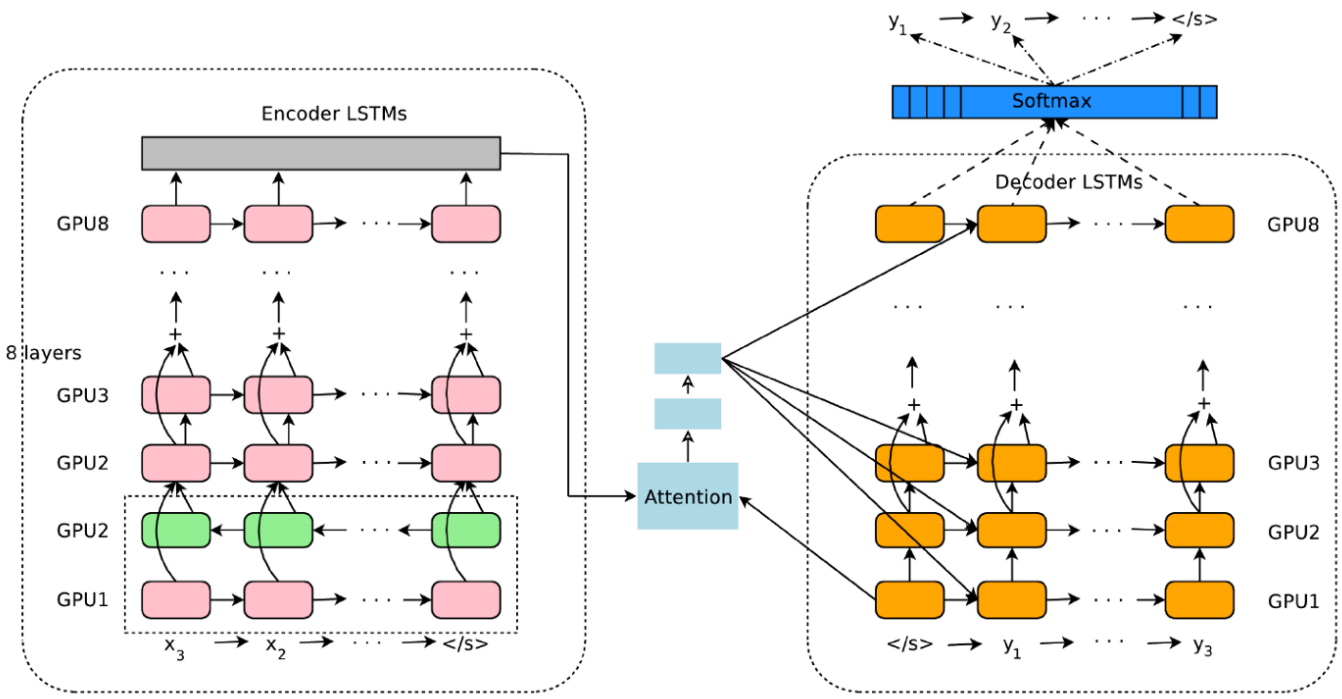
This is a paper in **2016 arXiv** with over **4800 citations**. (Sik-Ho Tsang @ Medium) There are many improvements in the paper as also seen that the author list is quite long with 31 authors. I try to make this article short.

In 2016 Sep, Google started to use Neural Machine Translation (NMT) instead of Statistical Machine Translation (SMT), which had been used since 2017 Oct.

Outline

3. Wordpiece Model or Mixed Word/Character Model
4. Model Training
5. Quantizable Model and Quantized Inference
6. Experimental Results

1. GNMT Network Architecture



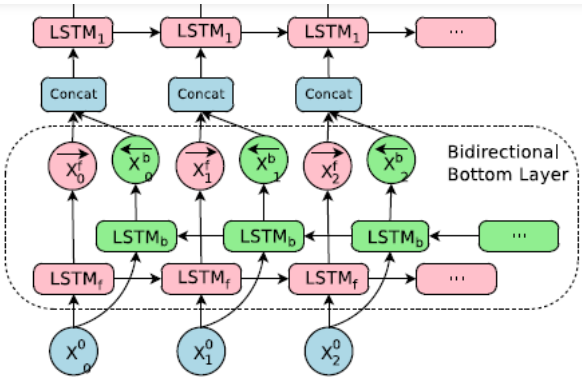
GNMT: Network Architecture

- On the **left** is the **encoder** network, on the **right** is the **decoder** network, in the **middle** is the **attention** module.
- Let (*X*, *Y*) be a source and target sentence pair. Let $X=x1, x2, x3, \ldots, xM$ be the sequence of *M* symbols in the source sentence and let $Y=y1, y2, y3, \ldots, yN$ be the sequence of *N* symbols in the target sentence.

$$\mathbf{x_1}, \mathbf{x_2}, ..., \mathbf{x_M} = EncoderRNN(x_1, x_2, x_3, ..., x_M)$$

- In this equation, $x1, x2, x3, \ldots, xM$ is a list of fixed size vectors.

1.1. Encoder



The structure of bi-directional connections in the first layer of the encoder

- The **bottom encoder** layer is **bi-directional**: the pink nodes gather information from left to right while the green nodes gather information from right to left.
- The **other** layers of the encoder are **uni-directional**.

1.2. Attention

> *The attention module is similar to the one in Attention Decoder.*

- More specifically, let ***yi-1*** be the **decoder-RNN output from the past decoding time step**. **Attention context *ai*** for the current time step is computed:

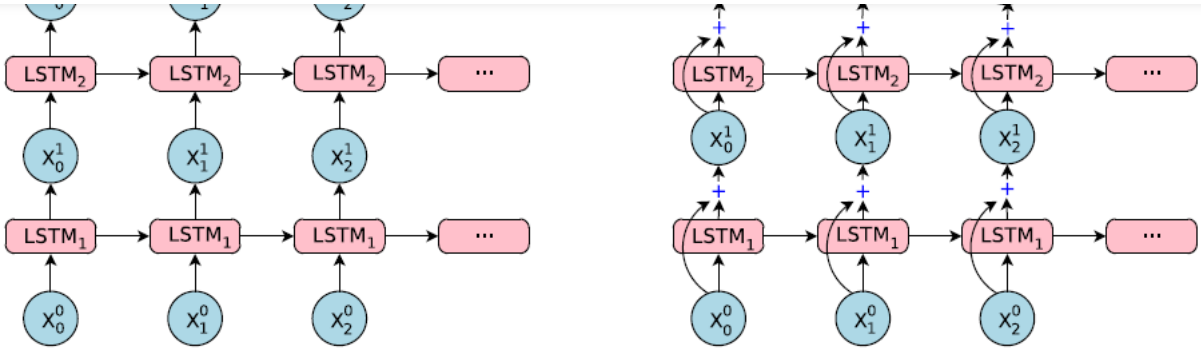
$$s_t = AttentionFunction(\mathbf{y}_{i-1}, \mathbf{x}_t) \quad \forall t, \quad 1 \le t \le M$$
$$p_t = \exp(s_t) / \sum_{t=1}^{M} \exp(s_t) \quad \forall t, \quad 1 \le t \le M$$
$$\mathbf{a}_i = \sum_{t=1}^{M} p_t . \mathbf{x}_t$$

- where **AttentionFunction** in our implementation is **a feed forward network with one hidden layer**.

1.3. Decoder

- **8 LSTM layers** are used for the decoder.
- **Beam search** is used during decoding to find the sequence *Y* that maximizes a score function *s*(*Y*, *X*) given a trained model.
- During beam search, 8–12 **hypotheses** are typically kept but it is found that using **fewer (4 or 2)** has only slight negative effects on BLEU scores.
- **Two important refinements** are used to the pure max-probability based beam search algorithm: a **coverage penalty** [42] and **length normalization.**
- This speeds up search by 30%-40% when run on CPUs.
- (Please feel free to read the paper for more details.)

1.4. Residual Connections



Left: Normal Stacked LSTM, Right: Stacked LSTM with Residual Connections

- **Left: Deep normal stacked LSTMs** are **difficult to train** due to exploding and vanishing gradient problems:

$$\begin{aligned}\mathbf{c}_t^i, \mathbf{m}_t^i &= \text{LSTM}_i(\mathbf{c}_{t-1}^i, \mathbf{m}_{t-1}^i, \mathbf{x}_t^{i-1}; \mathbf{W}^i) \\ \mathbf{x}_t^i &= \mathbf{m}_t^i \\ \mathbf{c}_t^{i+1}, \mathbf{m}_t^{i+1} &= \text{LSTM}_{i+1}(\mathbf{c}_{t-1}^{i+1}, \mathbf{m}_{t-1}^{i+1}, \mathbf{x}_t^i; \mathbf{W}^{i+1})\end{aligned}$$

- It is found that simple stacked LSTM layers **work well up to 4 layers, barely with 6 layers, and very poorly beyond 8 layers**.

> ***Right: Residual connections*** *greatly improve the gradient flow in the backward pass,* ***8 LSTM layers*** *are used for the encoder and decoder:*

$$\begin{aligned}\mathbf{c}_t^i, \mathbf{m}_t^i &= \text{LSTM}_i(\mathbf{c}_{t-1}^i, \mathbf{m}_{t-1}^i, \mathbf{x}_t^{i-1}; \mathbf{W}^i) \\ \mathbf{x}_t^i &= \mathbf{m}_t^i + \mathbf{x}_t^{i-1} \\ \mathbf{c}_t^{i+1}, \mathbf{m}_t^{i+1} &= \text{LSTM}_{i+1}(\mathbf{c}_{t-1}^{i+1}, \mathbf{m}_{t-1}^{i+1}, \mathbf{x}_t^i; \mathbf{W}^{i+1})\end{aligned}$$

- where $\boldsymbol{x}=\boldsymbol{m}+\boldsymbol{x}$ before going into next LSTM.

2. Model Parallelism

- The ***n* replicas** all share one copy of model parameters. n is often around 10. Each replica works on a mini-batch of m sentence pairs at a time, which is often 128 in the experiments.

> ***The encoder and decoder networks are partitioned along the depth dimension*** *and are* ***placed on multiple GPUs****, effectively running each layer on a different GPU, as shown in the figure of GNMT network architecture.*

- Since all but the first encoder layer are uni-directional, layer $i+1$ can start its computation before layer i is fully finished.

> *The* ***softmax*** *layer is also* ***partitioned****, with each partition responsible for a subset of symbols.*

3. Wordpiece Model or Mixed Word/Character Model

- There are two categories: **Word-based** and **character-based** models. In this work, GMNT proposes **wordpiece model**.
- To be brief, word-based model predicts word by word, and rare words are difficult to handle.
- Character-based model predicts character by character, the meaning of words is somehow lost.
- An example of turning words into wordpieces:

> *Wordpieces achieve a balance between the flexibility of characters and efficiency of words.*

- (Please feel free to read the paper for more details.)

4. Model Training

- The **standard maximum-likelihood (ML) training** objective is used:

$$\mathcal{O}_{\mathrm{ML}}(\boldsymbol{\theta}) = \sum_{i=1}^{N} \log P_\theta(Y^{*(i)} \mid X^{(i)}) \ .$$

- But this **does not directly improve the BLEU scores**.
- In brief, **model refinement** is done using the **expected reward objective** by **reinforcement learning (RL)**:

$$\mathcal{O}_{\mathrm{RL}}(\boldsymbol{\theta}) = \sum_{i=1}^{N} \sum_{Y \in \mathcal{Y}} P_\theta(Y \mid X^{(i)}) \ r(Y, Y^{*(i)}).$$

- where ***r*(*Y*, *Y* *(*i*))** denotes the **per-sentence score**, and we are computing an expectation over all of the output sentences *Y*, up to a certain length.
- To further stabilize training, **a linear combination of ML and RL** objectives is optimized as:

$$\mathcal{O}_{\mathrm{Mixed}}(\boldsymbol{\theta}) = \alpha * \mathcal{O}_{\mathrm{ML}}(\boldsymbol{\theta}) + \mathcal{O}_{\mathrm{RL}}(\boldsymbol{\theta})$$

- Due to the disadvantage of BLEU score, **GLEU score is proposed**.
- (Please feel free to read the paper for more details.)

5. Quantizable Model and Quantized Inference

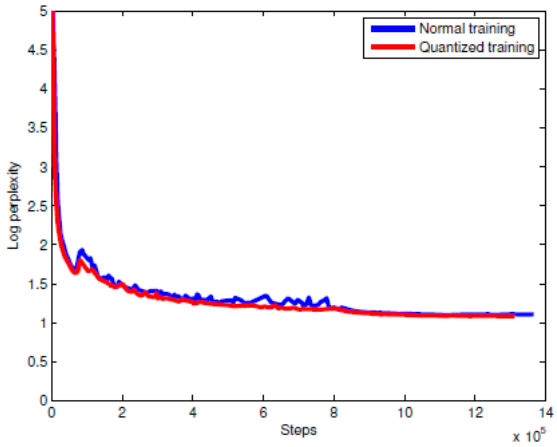
- Neural machine translation is **computationally intensive at inference**, making low latency translation difficult, and high volume deployment computationally expensive.
- **For quantized inference, GNMT explicitly constrains the values of these accumulators to be within [-δ∘, ∘δ]** to guarantee a certain range that can be used for quantization later. The **forward computation** of an LSTM stack with residual connections is as follows:

$$\begin{aligned}
\mathbf{c}'^{i}_t, \mathbf{m}^i_t &= \mathrm{LSTM}_i(\mathbf{c}^i_{t-1}, \mathbf{m}^i_{t-1}, \mathbf{x}^{i-1}_t; \mathbf{W}^i) \\
\mathbf{c}^i_t &= \max(-\delta, \min(\delta, \mathbf{c}'^{i}_t)) \\
\mathbf{x}'^{i}_t &= \mathbf{m}^i_t + \mathbf{x}^{i-1}_t \\
\mathbf{x}^i_t &= \max(-\delta, \min(\delta, \mathbf{x}'^{i}_t)) \\
\mathbf{c}'^{i+1}_t, \mathbf{m}^{i+1}_t &= \mathrm{LSTM}_{i+1}(\mathbf{c}^{i+1}_{t-1}, \mathbf{m}^{i+1}_{t-1}, \mathbf{x}^i_t; \mathbf{W}^{i+1}) \\
\mathbf{c}^{i+1}_t &= \max(-\delta, \min(\delta, \mathbf{c}'^{i+1}_t))
\end{aligned}$$

- The weights of fixed-point integer operations are replaced with **either 8-bit or 16-bit resolution**.
- There is also **softmax clipping *γ***:

$$\mathbf{p_t} = softmax(\mathbf{v'_t})$$

- During training of the model, full-precision is used. The only constraints added to the model during training are the clipping.



Log perplexity vs. steps

- And it is shown that it does not affect the training at all.

	BLEU	Log Perplexity	Decoding time (s)
CPU	31.20	1.4553	1322
GPU	31.20	1.4553	3028
TPU	31.21	1.4626	384

Model inference on CPU, GPU and TPU

- Inference using CPU is faster than the one in GPU due to data transfer.
- TPU is optimized which makes the inference much faster.
- (Please feel free to read the paper for more details.)

6. Experimental Results

6.1. ML Training Models

Model	BLEU	CPU decoding time per sentence (s)
Word	37.90	0.2226
Character	38.01	1.0530
WPM-8K	38.27	0.1919
WPM-16K	37.60	0.1874
WPM-32K	38.95	0.2118
Mixed Word/Character	38.39	0.2774
PBMT [15]	37.0	
LSTM (6 layers) [31]	31.5	
LSTM (6 layers + PosUnk) [31]	33.1	
Deep-Att [45]	37.7	
Deep-Att + PosUnk [45]	39.2	

Single model results on WMT En > Fr (newstest2014)

- **The best vocabulary size** for the mixed word-character model is **32K**.
- The best model WPM-32K, achieves a **BLEU score** of **38.95**. Note that this BLEU score represents the averaged score of 8 models. The maximum BLEU score of the 8 models is higher at 39.37.

Character (512 nodes)	22.62	0.8011
WPM-8K	23.50	0.2079
WPM-16K	24.36	0.1931
WPM-32K	24.61	0.1882
Mixed Word/Character	24.17	0.3268
PBMT [6]	20.7	
RNNSearch [37]	16.5	
RNNSearch-LV [37]	16.9	
RNNSearch-LV [37]	16.9	
Deep-Att [45]	20.6	

Single model results on WMT En > De (newstest2014)

- WMT En > De is considered a more difficult task than WMT En > Fr as it has **much less training data**.
- **It is more advantageous to use wordpiece or mixed word/character models**, which provide a gain of more than 2 BLEU points on top of the word model and about 4 BLEU points on top of previously reported results in [6] and Deep-Att [45].

6.2. RL Training Models

Dataset	Trained with log-likelihood	Refined with RL
En→Fr	38.95	39.92
En→De	24.67	24.60

Single model test BLEU scores, averaged over 8 runs

- On **WMT En > Fr, model refinement improves BLEU score by close to 1 point.**
- On **WMT En > De**, RL-refinement **slightly hurts** the test performance.

6.3. Model Ensemble and Human Evaluation

Model	BLEU
WPM-32K (8 models)	40.35
RL-refined WPM-32K (8 models)	41.16
LSTM (6 layers) [31]	35.6
LSTM (6 layers + PosUnk) [31]	37.5
Deep-Att + PosUnk (8 models) [45]	40.4

Model ensemble results on WMT En > Fr (newstest2014)

Model	BLEU
WPM-32K (8 models)	26.20
RL-refined WPM-32K (8 models)	26.30

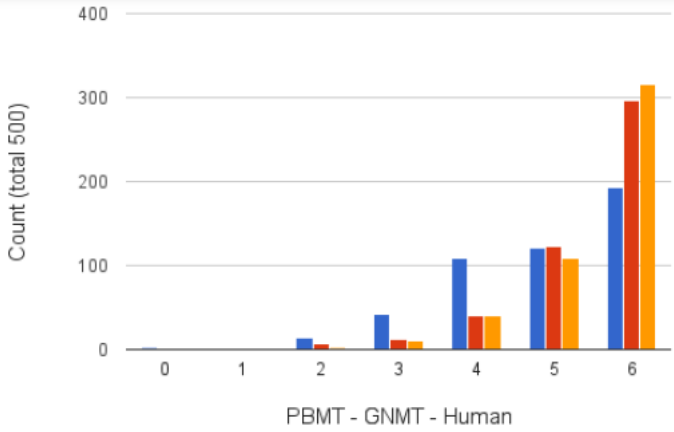
Model ensemble results on WMT En > De (newstest2014)

- GNMT **ensembles 8 RL-refined models** to obtain a state-of-the-art result of **41.16 BLEU** points on the **WMT En > Fr dataset**, outperforms Deep-Att.
- GNMT **ensembles 8 RL-refined models** to obtain a state-of-the-art result of **26.30 BLEU** points on the **WMT En > De dataset**.

Model	BLEU	Side-by-side averaged score
PBMT [15]	37.0	3.87
NMT before RL	40.35	4.46
NMT after RL	41.16	4.44
Human		4.82

Human side-by-side evaluation scores of WMT En > Fr models

- During the side-by-side comparison, humans are asked to rate four translations given a source sentence.
- Side-by-side scores range from 0 to 6, with a score of **0** meaning **"completely nonsense translation"**, and a score of **6** meaning **"perfect translation**.



Histogram of side-by-side scores on 500 sampled sentences from Wikipedia and news websites for a typical language pair, here English > Spanish (PBMT blue, GNMT red, Human orange)

	PBMT	GNMT	Human	Relative Improvement
English → Spanish	4.885	5.428	5.504	87%
English → French	4.932	5.295	5.496	64%
English → Chinese	4.035	4.594	4.987	58%
Spanish → English	4.872	5.187	5.372	63%
French → English	5.046	5.343	5.404	83%
Chinese → English	3.694	4.263	4.636	60%

Mean of side-by-side scores on production data

- **Google's translation production corpora are two to three decimal orders of magnitudes bigger** than the WMT corpora.
- **GMNT reduces translation errors by more than 60%** compared to the PBMT model on these major pairs of languages.

Reference

[2016 arXiv] [GNMT]
Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation

Natural Language Processing

Sequence Model: 2014 [GRU] [Doc2Vec]
Language Model: 2007 [Bengio TNN'07] **2013** [Word2Vec] [NCE] [Negative Sampling]
Sentence Embedding: 2015 [Skip-Thought]
Machine Translation: 2014 [Seq2Seq] [RNN Encoder-Decoder] **2015** [Attention Decoder/RNNSearch] **2016** [GNMT]
Image Captioning: 2015 [m-RNN] [R-CNN+BRNN] [Show and Tell/NIC]

My Other Previous Paper Readings