Parameter Sharing Methods for Multilingual Self-Attentional Translation Models

Devendra Sachan\textsuperscript{1} \hspace{1cm} Graham Neubig\textsuperscript{2}

\textsuperscript{1}Data Solutions Team, Petuum Inc, USA

\textsuperscript{2}Language Technologies Institute, Carnegie Mellon University, USA

Conference on Machine Translation, Nov 2018
Goal: Train a machine learning system to translate from multiple source languages to multiple target languages.
Goal: Train a machine learning system to translate from multiple source languages to multiple target languages.

Multilingual models follow the multi-task learning (MTL) paradigm.
Goal: Train a machine learning system to translate from multiple source languages to multiple target languages. Multilingual models follow the multi-task learning (MTL) paradigm

1. Models are jointly trained on data from several language pairs.
Goal: Train a machine learning system to translate from multiple source languages to multiple target languages.

Multilingual models follow the *multi-task learning* (MTL) paradigm

1. Models are jointly trained on data from several language pairs.
2. Incorporate some degree of parameter sharing.
One-to-Many Multilingual Translation

Translation from a common source language ("En") to multiple target languages ("De" and "Nl")
One-to-Many Multilingual Translation

- Translation from a common source language ("En") to multiple target languages ("De" and "Nl")
- Difficult task as we need to translate to (or generate) multiple target languages.
Previous Approach: Separate Decoders

- One shared encoder and one decoder per target language.¹

¹Multi-Task Learning for Multiple Language Translation, ACL 2015
Previous Approach: Separate Decoders

One shared encoder and one decoder per target language.\footnote{Multi-Task Learning for Multiple Language Translation, ACL 2015}

Advantage: ability to model each target language separately.
Previous Approach: Separate Decoders

- One shared encoder and one decoder per target language.\(^1\)
- Advantage: ability to model each target language separately.
- Disadvantages:
  1. Slower Training

\(^1\) Multi-Task Learning for Multiple Language Translation, ACL 2015
One shared encoder and one decoder per target language.\(^1\)

Advantage: ability to model each target language separately.

Disadvantages:
1. Slower Training
2. Increased memory requirements

\(^1\) Multi-Task Learning for Multiple Language Translation, ACL 2015
Previous Approach: Shared Decoder

▶ Single *unified* model: shared encoder and shared decoder for all language pairs.\(^2\)

---

\(^2\)Google’s Multilingual Neural Machine Translation System: Enabling Zero-Shot Translation, ACL 2017
Previous Approach: Shared Decoder

- Single *unified* model: shared encoder and shared decoder for all language pairs.\(^2\)

- Advantages:
  - Trivially implementable: using a standard bilingual translation model.

---

\(^2\)Google’s Multilingual Neural Machine Translation System: Enabling Zero-Shot Translation, ACL 2017
Previous Approach: Shared Decoder

- Single *unified* model: shared encoder and shared decoder for all language pairs.\(^2\)
- Advantages:
  - Trivially implementable: using a standard bilingual translation model.
  - Constant number of trainable parameters.

---

\(^2\)Google's Multilingual Neural Machine Translation System: Enabling Zero-Shot Translation, ACL 2017
Previous Approach: Shared Decoder

- Single *unified* model: shared encoder and shared decoder for all language pairs.\(^2\)
- Advantages:
  - Trivially implementable: using a standard bilingual translation model.
  - Constant number of trainable parameters.
- Disadvantage: decoder’s ability to model multiple languages can be significantly reduced.

\(^2\)Google’s Multilingual Neural Machine Translation System: Enabling Zero-Shot Translation, ACL 2017
Our Proposed Approach: **Partial Sharing**

- Share some but not all parameters.

![Diagram showing the proposed approach with a shared encoder and two decoders for different target languages.]
Our Proposed Approach: **Partial Sharing**

- Share **some but not all** parameters.
- Generalizes previous approaches.
Our Proposed Approach: Partial Sharing

- Share some **but not all** parameters.
- Generalizes previous approaches.
- We focus on the self-attentional Transformer model.
Transformer Model

3 Attention is all you need, NIPS 2017
Transformer Model\(^3\)

- **Embedding Layer**

---

\(^3\)Attention is all you need, NIPS 2017
Tasker Model

- Embedding Layer
- Encoder Layer (2 sublayers)

3 Attention is all you need, NIPS 2017
Transformer Model

- Embedding Layer
- Encoder Layer (2 sublayers)
  1. Self-attention

3Attention is all you need, NIPS 2017
Transformer Model

- Embedding Layer
- Encoder Layer (2 sublayers)
  1. Self-attention
  2. Feed-forward network

\(^3\text{Attention is all you need, NIPS 2017}\)
Transformer Model

- Embedding Layer
- Encoder Layer (2 sublayers)
  1. Self-attention
  2. Feed-forward network
- Decoder Layer (3 sublayers)

3 Attention is all you need, NIPS 2017
Transformer Model

- **Embedding Layer**
- **Encoder Layer (2 sublayers)**
  1. Self-attention
  2. Feed-forward network
- **Decoder Layer (3 sublayers)**
  1. Masked self-attention

---

3Attention is all you need, NIPS 2017
Transformer Model

- **Embedding Layer**
- **Encoder Layer (2 sublayers)**
  1. Self-attention
  2. Feed-forward network
- **Decoder Layer (3 sublayers)**
  1. Masked self-attention
  2. Encoder-decoder attention

\[^3\text{Attention is all you need, NIPS 2017}\]
Transformer Model

- **Embedding Layer**
- **Encoder Layer (2 sublayers)**
  1. Self-attention
  2. Feed-forward network
- **Decoder Layer (3 sublayers)**
  1. Masked self-attention
  2. Encoder-decoder attention
  3. Feed-forward network

---

Attention is all you need, NIPS 2017
Transformer Model

- Embedding Layer
- Encoder Layer (2 sublayers)
  1. Self-attention
  2. Feed-forward network
- Decoder Layer (3 sublayers)
  1. Masked self-attention
  2. Encoder-decoder attention
  3. Feed-forward network
- Output generation layer

---

\(^3\)Attention is all you need, NIPS 2017
Transformer Decoder’s Parameters

Embedding Layer

$W_E \in \mathbb{R}^{d_m \times V}$
Transformer Decoder’s Parameters

Embedding Layer

- $W_E \in \mathbb{R}^{d_m \times V}$

Masked Self-Attention

- $W^1_K, W^1_V, W^1_Q, W^1_F \in \mathbb{R}^{d_m \times d_m}$
Transformer Decoder’s Parameters

**Embedding Layer**
- \( W_E \in \mathbb{R}^{d_m \times V} \)

**Masked Self-Attention**
- \( W_K^1, W_V^1, W_Q^1, W_F^1 \in \mathbb{R}^{d_m \times d_m} \)

**Encoder-Decoder Attention**
- \( W_K^2, W_V^2, W_Q^2, W_F^2 \in \mathbb{R}^{d_m \times d_m} \)
Transformer Decoder’s Parameters

Embedding Layer

- $W_E \in \mathbb{R}^{d_m \times V}$

Masked Self-Attention

- $W^1_K, W^1_V, W^1_Q, W^1_F \in \mathbb{R}^{d_m \times d_m}$

Encoder-Decoder Attention

- $W^2_K, W^2_V, W^2_Q, W^2_F \in \mathbb{R}^{d_m \times d_m}$

Feed-Forward Network

- $W_{L1} \in \mathbb{R}^{d_m \times d_h}$
- $W_{L2} \in \mathbb{R}^{d_h \times d_m}$
Shareable parameters: embeddings, attention, embedding, linear layer weights.
Parameter Sharing Strategies

- $\Theta = \text{set of shared parameters}$
No Parameter Sharing

- Encoder 1 → Decoder 1
  Source Language: "En"
  Target Language: "De"

- Encoder 2 → Decoder 2
  Source Language: "En"
  Target Language: "Nl"

▶ Separate bilingual translation models
  \( \Theta = \emptyset \)
Embedding Sharing

- Common embedding layer
  \[ \Theta = \{ W_E \} \]
Encoder Sharing

- Common encoder and separate decoder for each target language

\[ \Theta = \{ W_E, \theta_{ENC} \} \]
Next, include decoder parameters among the set of shared parameters.
Next, include decoder parameters among the set of shared parameters.

Exponentially many combinations possible: only select a subset.
Decoder Sharing

- Next, include decoder parameters among the set of shared parameters.
- Exponentially many combinations possible: only select a subset.
- The selected weights are shared in all layers.
Parameter Sharing Strategies

**Encoder**
- Source Language: "En"
- Feed-Forward Network
- Self-Attention
- Embedding Layer

**Decoder 1**
- Target Language 1: "De"
- Feed-Forward Network
- Enc-Dec Attention
- Masked Self-Attention
- Embedding Layer
- Shareable Parameters
- Tied Linear Layer

**Decoder 2**
- Target Language 2: "Nl"
- Feed-Forward Network
- Enc-Dec Attention
- Masked Self-Attention
- Embedding Layer
- Tied Linear Layer

▶ **FFN sublayer parameters are shared**

$$\Theta = \{ W_E, \theta_{ENC}, W_{L1}, W_{L2} \}$$
Parameter Sharing Strategies

Sharing the weights of the self-attention sublayer

\[ \Theta = \{ W_E, \theta_{ENC}, W^K_1, W^Q_1, W^V_1, W^F_1 \} \]
Parameter Sharing Strategies

Sharing the weights of the encoder-decoder attention sublayer

\[ \Theta = \{ W_E, \theta_{ENC}, W_K^2, W_Q^2, W_V^2, W_F^2 \} \]
Limit the attention weights to the key and query weights

\[ \Theta = \{ W_E, \theta_{ENC}, W_K^1, W_Q^1, W_K^2, W_Q^2 \} \]
Parameter Sharing Strategies

- Limit the attention weights to the key and value weights

\[ \Theta = \{ W_E, \theta_{ENC}, W_K^1, W_V^1, W_K^2, W_V^2 \} \]
Parameter Sharing Strategies

Sharing all the decoder parameters to have a single unified model ($\Theta = \{W_E, \theta_{ENC}, \theta_{DEC}\}$)
Dataset

- Six language pairs from the TED talks dataset.\(^4\)
  https://github.com/neulab/word-embeddings-for-nmt

\(^4\)When and Why are Pre-trained Word Embeddings Useful for Neural Machine Translation?, NAACL 2018
Dataset

- Six language pairs from the TED talks dataset.\(^4\)
  https://github.com/neulab/word-embeddings-for-nmt
- Languages belong to different linguistic families

\(^4\)When and Why are Pre-trained Word Embeddings Useful for Neural Machine Translation?, NAACL 2018
Dataset

- Six language pairs from the TED talks dataset.\(^4\)  
  https://github.com/neulab/word-embeddings-for-nmt
- Languages belong to different linguistic families
  - Romanian (Ro) and French (Fr) are Romance languages

---

\(^4\)When and Why are Pre-trained Word Embeddings Useful for Neural Machine Translation?, NAACL 2018
Dataset

- Six language pairs from the TED talks dataset.  
  https://github.com/neulab/word-embeddings-for-nmt
- Languages belong to different linguistic families
  - Romanian (Ro) and French (Fr) are Romance languages
  - German (De) and Dutch (Nl) are Germanic languages

---

4 When and Why are Pre-trained Word Embeddings Useful for Neural Machine Translation?, NAACL 2018
Dataset

- Six language pairs from the TED talks dataset.\(^4\)
  - https://github.com/neulab/word-embeddings-for-nmt
- Languages belong to different linguistic families
  - Romanian (\texttt{Ro}) and French (\texttt{Fr}) are Romance languages
  - German (\texttt{De}) and Dutch (\texttt{Nl}) are Germanic languages
  - Turkish (\texttt{Tr}) and Japanese (\texttt{Ja}) are unrelated languages
    - Turkish: Turkic family
    - Japanese: Japonic family

\(^4\)When and Why are Pre-trained Word Embeddings Useful for Neural Machine Translation?, NAACL 2018
Multilingual Model Training Details

- Extra target language token at the start of source sentence.
Multilingual Model Training Details

- Extra target language token at the start of source sentence.
- Trained using balanced mini-batches for every target language.
Extra target language token at the start of source sentence.

Trained using balanced mini-batches for every target language.

Minimize weighted average cross-entropy loss.
Multilingual Model Training Details

- Extra target language token at the start of source sentence.
- Trained using balanced mini-batches for every target language.
- Minimize weighted average cross-entropy loss.
  - Weighting term is proportional to word count in target languages.
**Results**

**Baselines**

- **GNMT Model**: Based on recurrent LSTMs, residual connections, attention
Results

Baselines

- **GNMT Model**: Based on recurrent LSTMs, residual connections, attention
  1. **GNMT NS**: No Sharing
Results

Baselines

- **GNMT Model**: Based on recurrent LSTMs, residual connections, attention
  1. **GNMT NS**: No Sharing
  2. **GNMT FS**: Full Sharing
Results

Baselines

- **Transformer NS**: Separate models for each language pair
Results

Baselines

- **Transformer NS**: Separate models for each language pair
- **Transformer FS**: One model for all language pairs
Results: Target languages are from the same family
Results: Target languages are from the same family

BLEU Scores

▶ GNMT NS ≪ GNMT FS < TF NS ≪ TF FS
Results: Target languages are from different families
Results: Target languages are from different families

BLEU Scores

- **GNMT NS ≪ GNMT FS ≈ TF NS**
- **TF NS ≥ TF FS** for **En → De + Tr**
- **TF NS ≈ TF FS** for **En → De + Ja**
Results: Target languages are from the same family

Transformer Partial Sharing: $\Theta = \{W_E\}$

BLEU Scores:

- $\mathbf{TF \ FS} > \mathbf{TF \ PS}$ for $\text{En} \rightarrow \text{Ro} + \text{Fr}$
- $\mathbf{TF \ FS} \approx \mathbf{TF \ PS}$ for $\text{En} \rightarrow \text{De} + \text{Nl}$
Results: Target languages are from different families

Transformer Partial Sharing: $\Theta = \{W_E\}$

BLEU Scores

- $\text{TF FS} < \text{TF PS}$ for $\text{En} \rightarrow \text{De} + \text{Tr}$
- $\text{TF FS} \approx \text{TF PS}$ for $\text{En} \rightarrow \text{De} + \text{Ja}$
Results: Target languages are from the same family

Transformer Partial Sharing: $\Theta = \{W_E\} + \{\theta_{ENC}\}$

BLEU Scores:
- $\text{TF FS} > \text{TF PS}$ for $\text{En} \rightarrow \text{Ro} + \text{Fr}$ and $\text{En} \rightarrow \text{De} + \text{Nl}$
Results: Target languages are from different families

Transformer Partial Sharing: $\Theta = \{W_E\} + \{\theta_{ENC}\}$

BLEU Scores:

- $\text{TF FS} < \text{TF PS}$ for $\text{En} \rightarrow \text{De} + \text{Tr}$
- $\text{TF FS} \approx \text{TF PS}$ for $\text{En} \rightarrow \text{De} + \text{Ja}$
Results: Target languages are from the same family

Transformer Partial Sharing:
\[ \Theta = \{ W_E, \theta_{ENC} \} + \{ W_{L1}, W_{L2} \} \]

BLEU Scores:
- TF FS > TF PS for En → Ro + Fr and En → De + Nl
Results: Target languages are from different families

Transformer Partial Sharing:
\[ \Theta = \{ W_E, \theta_{ENC} \} + \{ W_{L1}, W_{L2} \} \]

BLEU Scores:
▶ TF FS < TF PS for En → De + Tr and En → De + Ja
Results: Target languages are from the same family

Transformer Partial Sharing:
\[ \Theta = \{ W_E, \theta_{ENC} \} + \{ W^1_K, W^1_Q, W^1_V, W^1_F \} \]

BLEU Scores:

- \( \text{TF FS} > \text{TF PS} \) for \( \text{En} \rightarrow \text{Ro + Fr} \)
- \( \text{TF FS} \approx \text{TF PS} \) for \( \text{En} \rightarrow \text{De + Nl} \)
Results: Target languages are from different families

Transformer Partial Sharing:
\[ \Theta = \{ W_E, \theta_{ENC} \} + \{ W^1_K, W^1_Q, W^1_V, W^1_F \} \]

BLEU Scores:
- \( \text{TF FS} < \text{TF PS} \) for En \( \rightarrow \) De + Tr
- \( \text{TF FS} \approx \text{TF PS} \) for En \( \rightarrow \) De + Ja
Results: Target languages are from the same family

Transformer Partial Sharing:
\[ \Theta = \{ W_E, \theta_{ENC} \} + \{ W_K^2, W_Q^2, W_V^2, W_F^2 \} \]

BLEU Scores:
- **TF FS \approx TF PS** for En \( \rightarrow \) Ro + Fr and En \( \rightarrow \) De + Nl
Results: Target languages are from different families

Transformer Partial Sharing:
\[ \Theta = \{W_E, \theta_{ENC}\} + \{W^2_K, W^2_Q, W^2_V, W^2_F\} \]

BLEU Scores:
- TF FS < TF PS for En → De + Tr
- TF FS ≈ TF PS for En → De + Ja
Results: Target languages are from the same family

Transformer Partial Sharing:
\[ \Theta = \{ W_E, \theta_{ENC} \} + \{ W_K^1, W_V^1, W_K^2, W_V^2 \} \]

BLEU Scores:
- TF FS > TF PS for En → Ro + Fr
- TF FS ≈ TF PS for En → De + Ni
Results: Target languages are from different families

Transformer Partial Sharing:
\[ \Theta = \{ W_E, \theta_{ENC} \} + \{ W^1_K, W^1_V, W^2_K, W^2_V \} \]

BLEU Scores:

▶ TF FS < TF PS for En → De + Tr and En → De + Ja
Results: Target languages are from the same family

Transformer Partial Sharing:
\[ \Theta = \{ W_E, \theta_{ENC} \} + \{ W_K^1, W_Q^1, W_K^2, W_Q^2 \} \]

BLEU Scores:
- TF FS ≈ TF PS for En \( \rightarrow \) Ro + Fr and En \( \rightarrow \) De + Nl
Results: Target languages are from different families

Transformer Partial Sharing:
\[ \Theta = \{ \mathbf{W}_E, \theta_{ENC} \} + \{ \mathbf{W}^1_K, \mathbf{W}^1_Q, \mathbf{W}^2_K, \mathbf{W}^2_Q \} \]

BLEU Scores:
- TF FS \ll TF PS for En \rightarrow De + Tr and En \rightarrow De + Ja
Results: Target languages are from the same family

▶ Sharing all parameters leads to the best BLEU scores for $\text{En} \rightarrow \text{Ro} + \text{Fr}$
Results: Target languages are from the same family

- Sharing all parameters leads to the best BLEU scores for $E_{N \rightarrow Ro+Fr}$
- Sharing only the key, query from both the decoder attention layers leads to the best BLEU scores for $E_{N \rightarrow De+Nl}$
Results: Target languages are from distant families

- Sharing all the parameters leads to a noticeable drop in the BLEU scores for both the considered language pairs.
Results: Target languages are from distant families

- Sharing all the parameters leads to a noticeable drop in the BLEU scores for both the considered language pairs.
- Sharing the key, query parameters results in a large increase in the BLEU scores.
Conclusions

- We explore parameter sharing strategies for multilingual translation using self-attentional models.
Conclusions

- We explore parameter sharing strategies for multilingual translation using self-attentional models.
- We examine the case when the target languages come from the same or distant language families.

Code: https://github.com/DevSinghSachan/multilingual-nmt
Conclusions

- We explore parameter sharing strategies for multilingual translation using self-attentional models.
- We examine the case when the target languages come from the same or distant language families.
- The popular approach of full parameter sharing may perform well only when the target languages belong to the same family.
- Partial parameter sharing of embedding, encoder, decoder's key, query weights is applicable to all kinds of language pairs.
- Partial parameter sharing achieves the best BLEU scores when the target languages are from distant families.

Code: https://github.com/DevSinghSachan/multilingual-nmt

Thank you! Questions?
Conclusions

- We explore parameter sharing strategies for multilingual translation using self-attentional models.
- We examine the case when the target languages come from the same or distant language families.
- The popular approach of full parameter sharing may perform well only when the target languages belong to the same family.
- Partial parameter sharing of embedding, encoder, decoder’s key, query weights is applicable to all kinds of language pairs.

Code: https://github.com/DevSinghSachan/multilingual

Thank you! Questions?
Conclusions

- We explore parameter sharing strategies for multilingual translation using self-attentional models.
- We examine the case when the target languages come from the same or distant language families.
- The popular approach of full parameter sharing may perform well only when the target languages belong to the same family.
- Partial parameter sharing of embedding, encoder, decoder’s key, query weights is applicable to all kinds of language pairs.
- Partial parameter sharing achieves the best BLEU scores when the target languages are from distant families.
Conclusions

isers

▶ We explore parameter sharing strategies for multilingual translation using self-attentional models.
▶ We examine the case when the target languages come from the same or distant language families.
▶ The popular approach of full parameter sharing may perform well only when the target languages belong to the same family.
▶ Partial parameter sharing of embedding, encoder, decoder’s key, query weights is applicable to all kinds of language pairs.
▶ Partial parameter sharing achieves the best BLEU scores when the target languages are from distant families.

Code: https://github.com/DevSinghSachan/multilingual_nmt
Conclusions

- We explore parameter sharing strategies for multilingual translation using self-attentional models.
- We examine the case when the target languages come from the same or distant language families.
- The popular approach of full parameter sharing may perform well only when the target languages belong to the same family.
- Partial parameter sharing of embedding, encoder, decoder’s key, query weights is applicable to all kinds of language pairs.
- Partial parameter sharing achieves the best BLEU scores when the target languages are from distant families.

Code: https://github.com/DevSinghSachan/multilingual_nmt

Thank you! Questions?