Pre-training Methods for Neural Machine Translation

Mingxuan Wang
ByteDance AI Lab

Lei Li
University of California, Santa Barbara
MT helps global information flow

7000 languages in the world

The unlabeled scripts of India are: (west) Gurmukhi, Gujarati, Kannada, Malayalam, and (east) Tamil, Telugu, Oriya, Bengali, Burmese.
Cross Language Barrier with Machine Translation

The latest version will launch in just a few months

Foreign Media

Global Conferences

Tourism

International Trade
Machine Translation has increased international trade by over 10%.

Equivalent to make the world smaller than 26%
Outline

• Basics
  – NMT
  – Pre-training paradigm

• Monolingual Pre-training for NMT
  – Pre-training style
  – Contrast to other data augmentation methods

• Multilingual Pre-training for NMT

• Pre-training for Speech Translation
PART I: Basics
What is Neural Machine Translation

Automatic conversion of text/speech from one natural language to another with a single neural network

French: Quand tu souris, le monde entier s’arrête et se fige un instant.

English: When you smile, the whole world stops and freezes for a moment.
I like singing and dancing.

我 喜欢 唱歌 和 跳舞.

Encoder-Decoder Paradigm

1. Encoding

2. Decoding
你好吗？

How are you?

Transformer Architecture

Encoder

Decoder

Add & Norm
Feed Forward
Add & Norm
Multi-Head Attention

Add & Norm
Feed Forward
Add & Norm
Multi-Head Attention

Beam Search

Multi-Head Attention
Masked Multi-Head Attention
Pre-training & Fine-tuning

Self-supervised learning without labels

Large, unlabelled data → Model → Pre-training task 1
Pre-training task 2
...
Pre-training task n

Fine-tune on downstream tasks

Small, labelled data → Model
Pre-training & Fine-tuning

Large, unlabelled data → Model → Pre-training task 1 → Pre-training task 2 → ... → Pre-training task n → Fine-tune on downstream tasks

Model → Model → Model → Model → Model → Model
Context Representations

• Semi-supervised sequence learning, Google 2015
Context Representations

- Elmo: Deep contextual word embeddings

Train Separate Left-to-Right and Right-to-Left LMs

Apply as “Pre-trained Embeddings”
Context Representations

- GPT: improve language understanding by generative pre-training

Train Deep (12-layer) Transformer LM

Fine-tune on Classification Task
Context Representations

- BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding
  - Bidirectional
  - Random mask

Unidirectional context
Build representation incrementally

Bidirectional context
Words can “see themselves”

store
the man went to the [MASK] to buy a [MASK] of milk

gallon

BERT: Pre-training and Fine-tuning

Pre-training

Fine-Tuning

Unlabeled Sentence A and B Pair

Question Answer Pair

Start/End Span

Masked Sentence A

Masked Sentence B
### GLUE Results

- **MultiNLI**
  - Premise: Hills and mountains are especially sanctified in Jainism.
  - Hypothesis: Jainism hates nature.
  - Label: Contradiction

- **CoLA**
  - Sentence: The wagon rumbled down the road.
  - Label: Acceptable
  - Sentence: The car honked down the road.
  - Label: Unacceptable

---

**BERT: Pre-training and Fine-tuning**

<table>
<thead>
<tr>
<th>System</th>
<th>MNLI-(m/mm)</th>
<th>QQP</th>
<th>QNLI</th>
<th>SST-2</th>
<th>CoLA</th>
<th>STS-B</th>
<th>MRPC</th>
<th>RTE</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>392k</td>
<td>363k</td>
<td>108k</td>
<td>67k</td>
<td>8.5k</td>
<td>5.7k</td>
<td>3.5k</td>
<td>2.5k</td>
<td></td>
</tr>
<tr>
<td>Pre-OpenAI SOTA</td>
<td>80.6/80.1</td>
<td>66.1</td>
<td>82.3</td>
<td>93.2</td>
<td>35.0</td>
<td>81.0</td>
<td>86.0</td>
<td>61.7</td>
<td>74.0</td>
</tr>
<tr>
<td>BiLSTM+ELMo+Attn</td>
<td>76.4/76.1</td>
<td>64.8</td>
<td>79.9</td>
<td>90.4</td>
<td>36.0</td>
<td>73.3</td>
<td>84.9</td>
<td>56.8</td>
<td>71.0</td>
</tr>
<tr>
<td>OpenAI GPT</td>
<td>82.1/81.4</td>
<td>70.3</td>
<td>88.1</td>
<td>91.3</td>
<td>45.4</td>
<td>80.0</td>
<td>82.3</td>
<td>56.0</td>
<td>75.2</td>
</tr>
<tr>
<td>BERT_BASE</td>
<td>84.6/83.4</td>
<td>71.2</td>
<td>90.1</td>
<td>93.5</td>
<td>52.1</td>
<td>85.8</td>
<td>88.9</td>
<td>66.4</td>
<td>79.6</td>
</tr>
<tr>
<td>BERT_LARGE</td>
<td><strong>86.7/85.9</strong></td>
<td><strong>72.1</strong></td>
<td><strong>91.1</strong></td>
<td><strong>94.9</strong></td>
<td><strong>60.5</strong></td>
<td><strong>86.5</strong></td>
<td><strong>89.3</strong></td>
<td><strong>70.1</strong></td>
<td><strong>81.9</strong></td>
</tr>
</tbody>
</table>

BERT achieves SOTA results on a huge number of NLP benchmarks.
Pre-training & Fine-tuning

Corpora/Data
- Youtube videos
- BooksCorpus
- Biomedical corpus
- Scientific publications
- SWAG, IMDB, Twitter
- Clinical notes/EHR
- Clinical notes/EHR
- Hierarchical diagnostic codes
- Monolingual Corpora (104 languages)

Graph Neural Net

Pre-trained models
- ScIBERT
- ERNIE(1)
- BioBERT
- BERT
- VideoBERT
- M-BERT
- ClinicalBERT
- ERNIE(2)
- TransBERT
- G-BERT

Pre-trained models
- BioBERT
- VideoBERT
- ScIBERT
- BERT
- ERNIE(1)
- ERNIE(2)
- ClinicalBERT
- TransBERT
- G-BERT

Fine-tuned models
- DocBERT
- Code Switching (e.g., English/Hindi mix sentences)
- Prediction tasks (e.g., Hospital readmission)
- Relation classification
- Classification tasks (e.g., medication recommendation)
- Video captioning (classification)
- Generic & Domain specific NLP tasks (e.g., NER)

Image Source
Pre-training & Fine-tuning

Does pre-training matter in NMT?
PART II: Monolingual Pre-training for NMT
Why Monolingual

MT: More data is better
Why Monolingual

MT: Parallel data is limited
PART2: Monolingual Pre-training for NMT

• The early stage
  – NMT initialized with word2vec [ACL 2017, NAACL 2018, AI 2020]
  – NMT initialized with language model [EMNLP 2017]

• BERT fusion
  – BERT Incorporating methods [ICLR 2020, AAAI 2020a]
  – BERT Tuning methods [AAAI 2020b]

• Unified sequence to sequence pre-training
  – MASS: Masked Sequence-to-Sequence Pre-training [ICML 2019]
  – BART: Denoising Sequence-to-Sequence Pre-training [ACL 2020]
• When and Why are Pre-trained Word Embeddings Useful for Neural Machine Translation [NAACL 2018]
• Improve Neural Machine Translation by Building Word Vector [AI 2020]
• A bag of useful tricks for practical neural machine translation: Embedding layer initialization and large batch size [ACL 2017]
• The pre-trained embeddings help more when the size of the training data is small
Effect of language similarity

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Lang. Family</th>
<th>std</th>
<th>pre</th>
</tr>
</thead>
<tbody>
<tr>
<td>ES → PT</td>
<td>West-Iberian</td>
<td>17.8</td>
<td>24.8 (+7.0)</td>
</tr>
<tr>
<td>FR → PT</td>
<td>Western Romance</td>
<td>12.4</td>
<td>18.1 (+5.7)</td>
</tr>
<tr>
<td>IT → PT</td>
<td>Romance</td>
<td>14.5</td>
<td>19.2 (+4.7)</td>
</tr>
<tr>
<td>RU → PT</td>
<td>Indo-European</td>
<td>2.4</td>
<td>8.6 (+6.2)</td>
</tr>
<tr>
<td>HE → PT</td>
<td>No Common</td>
<td>3.0</td>
<td>11.9 (+8.9)</td>
</tr>
</tbody>
</table>

- All pairs are trained on 40,000 sentences
- Language similarity with PT: ES>FR>IT>RU
  - BLEU improves: ES>FR>IT
- RU and HE have very low baseline BLEU scores, so it makes sense that their increases would be larger

Figure 1: BLEU and BLEU gain by data size.

That for all three languages the gain in BLEU score demonstrates a similar trend to that found in G

Table 3: Effect of linguistic similarity and pre-training

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Lang. Family</th>
<th>std</th>
<th>pre</th>
</tr>
</thead>
<tbody>
<tr>
<td>E</td>
<td>Indo-European</td>
<td>6.0</td>
<td>10.3 (+4.3)</td>
</tr>
<tr>
<td>R</td>
<td>Romance</td>
<td>10.7</td>
<td>15.0 (+4.3)</td>
</tr>
<tr>
<td>U</td>
<td>West-Iberian</td>
<td>4.0</td>
<td>6.3 (+2.3)</td>
</tr>
<tr>
<td>T</td>
<td>Portuguese</td>
<td>2.0</td>
<td>3.3 (+1.3)</td>
</tr>
</tbody>
</table>

When and Why are Pre-trained Word Embeddings Useful for Neural Machine Translation, [Qi et al NAACL 2018]
### Effect of language similarity

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Lang. Family</th>
<th>std</th>
<th>pre</th>
</tr>
</thead>
<tbody>
<tr>
<td>ES → PT</td>
<td>West-Iberian</td>
<td>17.8</td>
<td>24.8 (+7.0)</td>
</tr>
<tr>
<td>FR → PT</td>
<td>Western Romance</td>
<td>12.4</td>
<td>18.1 (+5.7)</td>
</tr>
<tr>
<td>IT → PT</td>
<td>Romance</td>
<td>14.5</td>
<td>19.2 (+4.7)</td>
</tr>
<tr>
<td>RU → PT</td>
<td>Indo-European</td>
<td>2.4</td>
<td>8.6 (+6.2)</td>
</tr>
<tr>
<td>HE → PT</td>
<td>No Common</td>
<td>3.0</td>
<td>11.9 (+8.9)</td>
</tr>
</tbody>
</table>

- All pairs are trained on 40,000 sentences
- Language similarity with PT: ES>FR>IT>RU
  - BLEU improves: ES>FR>IT
- RU and HE have very low baseline BLEU scores, so it makes sense that their increases would be larger
Effect of multilingual alignment

<table>
<thead>
<tr>
<th>Train</th>
<th>Eval</th>
<th>bi</th>
<th>std</th>
<th>pre</th>
<th>align</th>
</tr>
</thead>
<tbody>
<tr>
<td>GL + PT</td>
<td>GL</td>
<td>2.2</td>
<td>17.5</td>
<td>20.8</td>
<td>22.4</td>
</tr>
<tr>
<td>AZ + TR</td>
<td>AZ</td>
<td>1.3</td>
<td>5.4</td>
<td>5.9</td>
<td>7.5</td>
</tr>
<tr>
<td>BE + RU</td>
<td>BE</td>
<td>1.6</td>
<td>10.0</td>
<td>7.9</td>
<td>9.6</td>
</tr>
</tbody>
</table>

- Training on both low-resource and higher-resource languages, and test on only the low-resource language
  - bi: the bilingual baseline
  - std: the multilingual baseline
  - pre: pre-training word embedding
  - align: convert the word embeddings of multiple languages to a single space [Smith et al., 2017]

- Alignment ensures that the word embeddings of the two source languages are put into similar vector spaces, and improves the performance
• Unsupervised pretraining for sequence to sequence learning [EMNLP 2017]
• Exploiting Source-side Monolingual Data in Neural Machine Translation [EMNLP 2016]
• Semi-Supervised Learning for Neural Machine Translation [ACL 2016]
Unsupervised pretraining for sequence to sequence learning

- The red parameters are the encoder and the blue parameters are the decoder.
- All parameters in a shaded box are pre-trained with RNN language models.
- Otherwise, randomly initialized.

---

Unsupervised pretraining for sequence to sequence learning, [Ramachandran et al EMNLP 2017]
Pretraining on a lot of unlabeled data is essential. If the model is initialized with LMs that are pretrained on the source part and target part of the parallel corpus.
Unsupervised pretraining for sequence to sequence learning

Only pretraining the decoder is better than only pretraining the encoder.
Unsupervised pretraining for sequence to sequence learning

Pretrain as much as possible because the benefits compound.

Table 1: English to German performance on WMT test sets. Our pretrained model outperforms all other models. Note that the model without pretraining uses the LM objective.

Figure 3: English to German ablation study measuring the difference in validation BLEU between various ablations and the full model. More negative is worse. The full model uses LMs trained with monolingual data to initialize the encoder and decoder, plus the language modeling objective.

Results: Table 1 shows the results of our method in comparison with other baselines. Our method achieves a new state-of-the-art for single model performance on both newstest2014 and newstest2015, significantly outperforming the competitive semi-supervised backtranslation technique (Sennrich et al., 2015a). Equally impressive is the fact that our best single model outperforms the previous state of the art ensemble of 4 models. Our ensemble of 5 models matches or exceeds the previous best ensemble of 12 models.

Ablation study: In order to better understand the effects of pretraining, we conducted an ablation study by modifying the pretraining scheme. We were primarily interested in varying the pretraining scheme and the monolingual language modeling objectives because these two techniques produce the largest gains in the model. Figure 3 shows the drop in validation BLEU of various ablations compared with the full model. The full model uses LMs trained with monolingual data to initialize the encoder and decoder, in addition to the language modeling objective. In the follow...
Summary

• Insight
  – Pre-training is effective on low-resource NMT
  – Pre-training as much as components
  – Pre-training as much as training data
  – Cross-lingual information helps

• Limitations:
  – The improvements on rich resource NMT is not large enough
  – The pre-training model is trained on limited training corpus, e.g. the monolingual part of the parallel data
  – Only a subset of parameters are pre-trained
Then, BERT comes…
What happens?

Pre-training data scale increased

![Graph showing the increase in monolingual data scale for Semi-Pre, Seq-pre, and BERT models. BERT model shows the highest increase.](image-url)
What happens?

Pre-training framework changed

![Diagram showing pre-training framework change]
What happens?

Baseline improved

WMT14 En-De

- GNMT14
- Semi-Pre16
- Transformer-base17
- Transformer-big17
Does BERT matter in NMT?

What happens?
PART2: Monolingual Pre-training for NMT

- The Bronze Age
  - NMT initialized with word2vec [ACL 2017, NAACL 2018, AI 2020]
  - NMT initialized with language model [EMNLP 2017]

- BERT fusion
  - BERT Incorporating methods [ICLR 2020, AAAI 2020a]
  - BERT Tuning methods [AAAI 2020b]

- Unified sequence to sequence pre-training
  - MASS: Masked Sequence-to-Sequence Pre-training [ICML 2019]
  - BART: Denoising Sequence-to-Sequence Pre-training [ACL 2020]
Fine-tuning BERT does **NOT** work!
- BERT and XLM pre-training for the encoder decreased the performance
- XLM pre-training for the decoder enlarged the performance gap

**BERT-Frozen achieved improvements**

---

### Table 1: Preliminary explorations on IWSLT’14 English→German translation

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>BLEU score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standard Transformer</td>
<td>28.57</td>
</tr>
<tr>
<td>Use BERT to initialize the encoder of NMT</td>
<td>27.14</td>
</tr>
<tr>
<td>Use XLM to initialize the encoder of NMT</td>
<td>28.22</td>
</tr>
<tr>
<td>Use XLM to initialize the decoder of NMT</td>
<td>26.13</td>
</tr>
<tr>
<td>Use XLM to initialize both the encoder and decoder of NMT</td>
<td>28.99</td>
</tr>
<tr>
<td>Leveraging the output of BERT as embeddings</td>
<td>29.67</td>
</tr>
</tbody>
</table>
Incorporate BERT into Neural Machine Translation

• BERT features are directly fed to both encoder and decoder layers
• Additional attention model to incorporate BERT features

Incorporate BERT into Neural Machine Translation, [Zhu et al ICLR 2020]
Datasets and settings

• Fine-tuning dataset
  – Low resource: IWSLT En-De, En-FR, En-Zh, En-Es (less than 250 k sentence pairs)
  – Rich resource: WMT14 En-De and En-Fr (4 M and 36 M sentence pairs)

• Settings
  – BERT base for IWSLT
  – BERT large for WMT
  – Both the BERT-encoder and BERTdecoder attention are randomly initialized

Incorporate BERT into Neural Machine Translation, [Zhu et al ICLR 2020]
Main results on supervised MT

- Experiments on a strong baseline
- BERT-fused model outperforms transformer baseline in all settings

Incorporate BERT into Neural Machine Translation, [Zhu et al ICLR 2020]
• Pre-training plays an crucial role in unsupervised NMT (Lample v.s. xml, mass and BERT-fused)
• BERT-fused outperforms XLM and MASS
• The comparison is slightly unfair, since BERT-fused introduced additional parameters
Jointly train BERT model with the NMT can also boost the baseline from 28.57 to 28.87.
But it is not as good as fixing the BERT part, whose BLEU is 30.45.

Table 6: Ablation study on IWSLT’14 En→De.

<table>
<thead>
<tr>
<th>Method</th>
<th>BLEU</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standard Transformer</td>
<td>28.57</td>
</tr>
<tr>
<td>BERT-fused model</td>
<td>30.45</td>
</tr>
<tr>
<td>Randomly initialize encoder/decoder of BERT-fused model</td>
<td>27.03</td>
</tr>
<tr>
<td>Jointly tune BERT and encoder/decoder of BERT-fused model</td>
<td>28.87</td>
</tr>
<tr>
<td>Feed BERT feature into all layers without attention</td>
<td>29.61</td>
</tr>
<tr>
<td>Replace BERT output with random vectors</td>
<td>28.91</td>
</tr>
<tr>
<td>Replace BERT with the encoder of another Transformer model</td>
<td>28.99</td>
</tr>
<tr>
<td>Remove BERT-encoder attention</td>
<td>29.87</td>
</tr>
<tr>
<td>Remove BERT-decoder attention</td>
<td>29.90</td>
</tr>
</tbody>
</table>

Incorporate BERT into Neural Machine Translation, [Zhu et al ICLR 2020]
NMT pre-training is also important to the success of BERT-fused model. Without NMT pre-training, the performance lags behind the baseline model.

Table 6: Ablation study on IWSLT’14 En→De.

<table>
<thead>
<tr>
<th>Method</th>
<th>BLEU</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standard Transformer</td>
<td>28.57</td>
</tr>
<tr>
<td>BERT-fused model</td>
<td>30.45</td>
</tr>
<tr>
<td>Randomly initialize encoder/decoder of BERT-fused model</td>
<td>27.03</td>
</tr>
<tr>
<td>Jointly tune BERT and encoder/decoder of BERT-fused model</td>
<td>28.87</td>
</tr>
<tr>
<td>Feed BERT feature into all layers without attention</td>
<td>29.61</td>
</tr>
<tr>
<td>Replace BERT output with random vectors</td>
<td>28.91</td>
</tr>
<tr>
<td>Replace BERT with the encoder of another Transformer model</td>
<td>28.99</td>
</tr>
<tr>
<td>Remove BERT-encoder attention</td>
<td>29.87</td>
</tr>
<tr>
<td>Remove BERT-decoder attention</td>
<td>29.90</td>
</tr>
</tbody>
</table>

NMT Pre-training is also important to the success of BERT-fused model. Without NMT pre-training, the performance lags behind the baseline model.
Remove attention module, the performance still outperforms baseline, but falls behind BERT-fused model.

It suggests that separate BERT model provides additional gains.

Table 6: Ablation study on IWSLT’14 En→De.

<table>
<thead>
<tr>
<th>Method</th>
<th>BLEU</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standard Transformer</td>
<td>28.57</td>
</tr>
<tr>
<td>BERT-fused model</td>
<td>30.45</td>
</tr>
<tr>
<td>Randomly initialize encoder/decoder of BERT-fused model</td>
<td>27.03</td>
</tr>
<tr>
<td>Jointly tune BERT and encoder/decoder of BERT-fused model</td>
<td>28.87</td>
</tr>
<tr>
<td><strong>Feed BERT feature into all layers without attention</strong></td>
<td>29.61</td>
</tr>
<tr>
<td>Replace BERT output with random vectors</td>
<td>28.91</td>
</tr>
<tr>
<td>Replace BERT with the encoder of another Transformer model</td>
<td>28.99</td>
</tr>
<tr>
<td>Remove BERT-encoder attention</td>
<td>29.87</td>
</tr>
<tr>
<td>Remove BERT-decoder attention</td>
<td>29.90</td>
</tr>
</tbody>
</table>

Incorporate BERT into Neural Machine Translation, [Zhu et al ICLR 2020]
Of course, BERT matters

Replace BERT representation with another transformer model, the performance drops significantly. It indicates BERT provides meaningful information and the improvements is not from the additional parameters.

Table 6: Ablation study on IWSLT’14 En→De.

<table>
<thead>
<tr>
<th>Method</th>
<th>BLEU</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standard Transformer</td>
<td>28.57</td>
</tr>
<tr>
<td>BERT-fused model</td>
<td>30.45</td>
</tr>
<tr>
<td>Randomly initialize encoder/decoder of BERT-fused model</td>
<td>27.03</td>
</tr>
<tr>
<td>Jointly tune BERT and encoder/decoder of BERT-fused model</td>
<td>28.87</td>
</tr>
<tr>
<td>Feed BERT feature into all layers without attention</td>
<td>29.61</td>
</tr>
<tr>
<td>Replace BERT output with random vectors</td>
<td>28.91</td>
</tr>
<tr>
<td>Replace BERT with the encoder of another Transformer model</td>
<td>28.99</td>
</tr>
<tr>
<td>Remove BERT-encoder attention</td>
<td>29.87</td>
</tr>
<tr>
<td>Remove BERT-decoder attention</td>
<td>29.90</td>
</tr>
</tbody>
</table>

Incorporate BERT into Neural Machine Translation, [Zhu et al ICLR 2020]
Acquiring Knowledge from Pre-trained Model to Neural Machine Translation

• Key idea
  – Dynamic fusion of different BERT layers, while BERT-fused model only uses the last layer of BERT
  – Incorporate BERT into all encoder layers and decoder layers with adaptive weight
  – Experiments including both BERT & GPT
We implement our approach with the in-house implementation of the Transformer (Dou et al. 2018; Vaswani et al. 2017). We use beam search for beam size 4. We measure the translation quality with the NIST-BLEU (Papineni et al. 2002). We also implement GPT (Radford et al. 2018), BERT (Devlin et al. 2018) and MASS (Song et al. 2019) in our Transformer system.

The Transformer settings of Transformer follow Vaswani et al. (2017). Each batch has 50 sentence and the maximum length of a sentence is limited to 100. We use label smoothing with value 0.1. The number of layers for the encoder and decoder are 6. Sentence pairs are batched together by approximate sentence length. The initialization and fine-tuning method with different pre-trained models are reported in Table 1.

After the training stage, we use beam search for heuristics, and the rest of the sentence as the input of encoder. It masks a continuous segment from a sentence as the label, and the rest of the sentence as the input of encoder. The objective is limited to 100. We use label smoothing with value 0.1. The number of layers for the encoder and decoder are 6. Sentence pairs are batched together by approximate sentence length. Each batch has 50 sentence and the maximum length of a sentence is limited to 100. We use label smoothing with value 0.1.

### Table 1: Translation qualities on the EN

<table>
<thead>
<tr>
<th>Model</th>
<th>Pre-trained Model</th>
<th>EN→DE BLEU</th>
<th>DE→EN BLEU</th>
<th>ZH→EN BLEU</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transformer (Vaswani et al. 2017)</td>
<td>N/A N/A</td>
<td>27.3</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Transformer (Zheng et al. 2019)</td>
<td>N/A N/A</td>
<td>27.14</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Transformer (Dou et al. 2018)</td>
<td>N/A N/A</td>
<td>27.31</td>
<td>N/A</td>
<td>24.13</td>
</tr>
<tr>
<td>Transformer</td>
<td>N/A N/A</td>
<td>27.31</td>
<td>32.51</td>
<td>24.47</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>w/ Fine-tuning</th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Transformer w/ Fine-tuning</td>
<td>GPT N/A</td>
<td>27.82 (+0.51)</td>
<td>33.17 (+0.66)</td>
<td>25.11 (+0.64)</td>
</tr>
<tr>
<td></td>
<td>N/A GPT</td>
<td>27.45 (+0.14)</td>
<td>32.87 (+0.36)</td>
<td>24.59 (+0.12)</td>
</tr>
<tr>
<td></td>
<td>GPT GPT</td>
<td>27.85 (+0.54)</td>
<td>32.79 (+0.28)</td>
<td>25.21 (+0.74)</td>
</tr>
<tr>
<td></td>
<td>BERT N/A</td>
<td>28.22 (+0.91)</td>
<td>33.64 (+1.13)</td>
<td>25.33 (+0.86)</td>
</tr>
<tr>
<td></td>
<td>N/A BERT</td>
<td>27.42 (+0.11)</td>
<td>33.13 (+0.62)</td>
<td>24.78 (+0.31)</td>
</tr>
<tr>
<td></td>
<td>BERT BERT</td>
<td>28.32 (+1.01)</td>
<td>33.57 (+1.06)</td>
<td>25.45 (+0.98)</td>
</tr>
<tr>
<td></td>
<td>GPT BERT</td>
<td>28.29 (+0.98)</td>
<td>33.33 (+0.82)</td>
<td>25.42 (+0.95)</td>
</tr>
<tr>
<td></td>
<td>BERT GPT</td>
<td>28.32 (+1.01)</td>
<td>33.57 (+1.05)</td>
<td>25.46 (+0.99)</td>
</tr>
<tr>
<td></td>
<td>MASS DAE</td>
<td>28.07 (+0.76)</td>
<td>33.29 (+0.78)</td>
<td>25.11 (+0.64)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>27.63 (+0.33)</td>
<td>33.03 (+0.52)</td>
<td>24.67 (+0.20)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>w/ APT Framework</td>
<td>GPT BERT</td>
<td>28.89 (+1.58)</td>
<td>34.32 (+1.81)</td>
<td>25.98 (+1.51)</td>
</tr>
<tr>
<td></td>
<td>BERT GPT</td>
<td><strong>29.23 (+1.92)</strong></td>
<td><strong>34.84 (+2.33)</strong></td>
<td><strong>26.21 (+1.74)</strong></td>
</tr>
<tr>
<td></td>
<td>GPT GPT</td>
<td>28.97 (+1.66)</td>
<td>34.26 (+1.75)</td>
<td>26.01 (+1.54)</td>
</tr>
<tr>
<td></td>
<td>BERT BERT</td>
<td>29.02 (+1.71)</td>
<td>34.67 (+2.16)</td>
<td><strong>26.46 (+1.99)</strong></td>
</tr>
</tbody>
</table>

**Table 1:** Translation qualities on the EN and ZH

<table>
<thead>
<tr>
<th>Model</th>
<th>Pre-trained Model</th>
<th>EN→DE BLEU</th>
<th>DE→EN BLEU</th>
<th>ZH→EN BLEU</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transformer (Vaswani et al. 2017)</td>
<td>N/A N/A</td>
<td>27.3</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Transformer (Zheng et al. 2019)</td>
<td>N/A N/A</td>
<td>27.14</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Transformer (Dou et al. 2018)</td>
<td>N/A N/A</td>
<td>27.31</td>
<td>N/A</td>
<td>24.13</td>
</tr>
<tr>
<td>Transformer</td>
<td>N/A N/A</td>
<td>27.31</td>
<td>32.51</td>
<td>24.47</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>w/ Fine-tuning</th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Transformer w/ Fine-tuning</td>
<td>GPT N/A</td>
<td>27.82 (+0.51)</td>
<td>33.17 (+0.66)</td>
<td>25.11 (+0.64)</td>
</tr>
<tr>
<td></td>
<td>N/A GPT</td>
<td>27.45 (+0.14)</td>
<td>32.87 (+0.36)</td>
<td>24.59 (+0.12)</td>
</tr>
<tr>
<td></td>
<td>GPT GPT</td>
<td>27.85 (+0.54)</td>
<td>32.79 (+0.28)</td>
<td>25.21 (+0.74)</td>
</tr>
<tr>
<td></td>
<td>BERT N/A</td>
<td>28.22 (+0.91)</td>
<td>33.64 (+1.13)</td>
<td>25.33 (+0.86)</td>
</tr>
<tr>
<td></td>
<td>N/A BERT</td>
<td>27.42 (+0.11)</td>
<td>33.13 (+0.62)</td>
<td>24.78 (+0.31)</td>
</tr>
<tr>
<td></td>
<td>BERT BERT</td>
<td>28.32 (+1.01)</td>
<td>33.57 (+1.06)</td>
<td>25.45 (+0.98)</td>
</tr>
<tr>
<td></td>
<td>GPT BERT</td>
<td>28.29 (+0.98)</td>
<td>33.33 (+0.82)</td>
<td>25.42 (+0.95)</td>
</tr>
<tr>
<td></td>
<td>BERT GPT</td>
<td>28.32 (+1.01)</td>
<td>33.57 (+1.05)</td>
<td>25.46 (+0.99)</td>
</tr>
<tr>
<td></td>
<td>MASS DAE</td>
<td>28.07 (+0.76)</td>
<td>33.29 (+0.78)</td>
<td>25.11 (+0.64)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>27.63 (+0.33)</td>
<td>33.03 (+0.52)</td>
<td>24.67 (+0.20)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>w/ APT Framework</td>
<td>GPT BERT</td>
<td>28.89 (+1.58)</td>
<td>34.32 (+1.81)</td>
<td>25.98 (+1.51)</td>
</tr>
<tr>
<td></td>
<td>BERT GPT</td>
<td><strong>29.23 (+1.92)</strong></td>
<td><strong>34.84 (+2.33)</strong></td>
<td><strong>26.21 (+1.74)</strong></td>
</tr>
<tr>
<td></td>
<td>GPT GPT</td>
<td>28.97 (+1.66)</td>
<td>34.26 (+1.75)</td>
<td>26.01 (+1.54)</td>
</tr>
<tr>
<td></td>
<td>BERT BERT</td>
<td>29.02 (+1.71)</td>
<td>34.67 (+2.16)</td>
<td><strong>26.46 (+1.99)</strong></td>
</tr>
</tbody>
</table>

**Table 1:** Translation qualities on the EN and ZH

**GPT v.s. BERT**

Although our work combining with GPT gets better performance than BERT due to it can obtain more contextual information. While on the decoder language model of GPT, the masked language model could mask a continuous segment from a sentence as the label, and the rest of the sentence as the input of encoder. First, BERT is better than GPT on the encoder when the fine-tuning method with different pre-trained models. When the encoder is initialized by BERT and the decoder is initialized by BERT or GPT, the BLEU score improves about 1 point percentage improvement on the ZH experiments.
We use the which tests a model’s ability to discriminate be-
and BERT-fused model as reported in
We successfully reproduced the BLUE scores of the baseline
dumps.wikimedia.org/dewiki/latest
fused model demonstrates stronger performances
For both rich- and low-resource settings, the BERT-
percentage of correct decisions as results.
the decision as correct. We aggregate model deci-
tences. If the model score of the actual translation
models can score a negative log probability for sen-
indicating the difference in performance between
between masculine and feminine scores, and
served the gender of the entity from the original sen-
3 aspects: overall accuracy calculated by the per-
bias is evaluated on morphological analysis from
and idiom translation are evaluated on the presence
4.4 Evaluation
monolingual sentences in English. All datasets are
therefore we choose a suitable amount and scale up
monolingual data from the same source of BERT
Mathis et al.
Pre-training has better generalization ability

<table>
<thead>
<tr>
<th>System</th>
<th>En→De</th>
<th>Zh→En</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standard Transformer</td>
<td>29.20</td>
<td>45.15</td>
</tr>
<tr>
<td>+ back translation (1:0.5)</td>
<td><strong>30.41</strong></td>
<td>46.70</td>
</tr>
<tr>
<td>+ back translation (1:1)</td>
<td>30.25</td>
<td><strong>47.23</strong></td>
</tr>
<tr>
<td>+ back translation (1:2)</td>
<td>30.18</td>
<td>47.04</td>
</tr>
<tr>
<td>+ back translation (1:4)</td>
<td>30.25</td>
<td>46.39</td>
</tr>
<tr>
<td>BERT-fused model</td>
<td>30.03</td>
<td>46.55</td>
</tr>
</tbody>
</table>

- Pre-training is much more promising
  - better generalization ability
  - Back translation is limited with data scale

Comparison between Pre-training and Large-scale Back-translation, [Huang et al ACL 2021]
Summary

• Advantages
  – BERT features are fused in all layers
  – Additional attention model adaptively determine how to leverage BERT feature

• Limitations
  – Additional cost including training storage and inference time
  – Why not tune BERT?
Towards Making Most of BERT for NMT

Why simply incorporating BERT does not work as expectation

• Fine-tuning leads to performance degradation on the original task
• The situation is more severe on NMT fine-tuning
  • High capacity of baseline needs much updating
  • Updating too much makes the model forgets its universal knowledge from pre-training
Not tuning too much

- Concerted training framework
  - Rate-scheduled Learning
  - Dynamic Switch
  - Asymptotic Distillation

Towards Making Most of BERT for NMT, [Yang et al AAAI 2020]
Rate-scheduled Learning rate

- Gradually increase the learning rate of BERT parameters from 0 to 1
- Then, decrease the learning rate of BERT parameters from 1 to 0
- Keep the BERT parameters frozen

Rate-scheduled learning rate is actually a trade off between fine-tuning and BERT frozen
Not tuning too much

• Dynamic Switch
  • Use a gate to dynamically decide which part is more important
  • If $\sigma$ is learned to 0, it degrade to the NMT model
  • If $\sigma$ is learned to 1, it simply act as Bert fine-tune approach

Dynamic Switch is more flexible than rate-scheduled learning rate
Not tuning too much

• Asymptotic Distillation
  • The pre-trained BERT serves as a teacher network while the encoder of the NMT model serves as a student
  • Minimize MSE loss of hidden states between NMT encoder and BERT to retain the pre-trained information
  • Use a hyper-parameter to balance the preference between pre-training distillation and NMT objective

Distillation Without introducing of additional parameters!

Towards Making Most of BERT for NMT, [Yang et al AAAI 2020]
### Towards Making Most of BERT for NMT, [Yang et al AAAI 2020]

<table>
<thead>
<tr>
<th>System</th>
<th>Architecture</th>
<th>En-De</th>
<th>En-Fr</th>
<th>En-Zh</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Existing systems</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Vaswani et al. (2017)</td>
<td>Transformer base</td>
<td>27.3</td>
<td>38.1</td>
<td>-</td>
</tr>
<tr>
<td>Vaswani et al. (2017)</td>
<td>Transformer big</td>
<td>28.4</td>
<td>41.0</td>
<td>-</td>
</tr>
<tr>
<td>Lample and Conneau (2019)</td>
<td>Transformer big + Fine-tuning</td>
<td>27.7</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Lample and Conneau (2019)</td>
<td>Transformer big + Frozen Feature</td>
<td>28.7</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Chen et al. (2018)</td>
<td>RNMT+ + MultiCol</td>
<td>28.7</td>
<td>41.7</td>
<td>-</td>
</tr>
<tr>
<td><strong>Our NMT systems</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CTNMT</td>
<td>Transformer (base)</td>
<td>27.2</td>
<td>41.0</td>
<td>37.3</td>
</tr>
<tr>
<td>CTNMT</td>
<td>Rate-scheduling</td>
<td>29.7</td>
<td>41.6</td>
<td>38.4</td>
</tr>
<tr>
<td>CTNMT</td>
<td>Dynamic Switch</td>
<td>29.4</td>
<td>41.4</td>
<td>38.6</td>
</tr>
<tr>
<td>CTNMT</td>
<td>Asymptotic Distillation</td>
<td>29.2</td>
<td>41.6</td>
<td>38.3</td>
</tr>
<tr>
<td><strong>CTNMT</strong></td>
<td>+ ALL</td>
<td><strong>30.1</strong></td>
<td><strong>42.3</strong></td>
<td><strong>38.9</strong></td>
</tr>
</tbody>
</table>

- Three strategies can independently work well on WMT14 En-De, En-Fr and WMT18 En-Zh
- CTNMT base model achieves even better results than Transformer big model
CTNMT outperforms fine-tuning on all training steps.
The performance gaps is enlarged as the fine-tuning steps increasing.
Summary

• Advantage
  – Simple and effective, obtains +3 BLEU on WMT14 en-de benchmark
  – Three methods can be used separately or jointly

• Limitation
  – Introducing pre-training method for decoder is promising but still difficult
  – Cross attention is important but not pre-trained

<table>
<thead>
<tr>
<th>Models</th>
<th>En→De BLEU</th>
</tr>
</thead>
<tbody>
<tr>
<td>BERT Enc</td>
<td>29.2</td>
</tr>
<tr>
<td>BERT Dec</td>
<td>26.1</td>
</tr>
<tr>
<td>GPT-2 Enc</td>
<td>27.7</td>
</tr>
<tr>
<td>GPT-2 Dec</td>
<td>27.4</td>
</tr>
</tbody>
</table>

Table 1: Case-sensitive BLEU scores on English-German, English-French and English-Chinese translation. The best performance comes from the fusion of rate-scheduling, dynamic switch and asymptotic distillation. Performance. And with only Asymptotic Distillation we still outperform MultiCol without additional parameters.

4 Results and Analysis
The results on English-German and English-French translation are presented in Table 1. We compare CT NMT with various other systems including Transformer and previous state-of-the-art pre-trained LM enhanced model. As observed by Edunov et al. (2019), Transformer big model with fine-tuning approach even falls behind the baseline. They then freeze the LM parameters during fine-tuning and achieve a few gains over the strong transformer big model. This is consistent with our intuition that fine-tuning on the large dataset may lead to degradation of the performance. In CT NMT, we first evaluate the effectiveness of the proposed three strategies respectively. Clearly, these methods achieve almost 2 BLEU score improvement over the state-of-the-art on the English-German task for the base network. In the case of the larger English-French task, we obtain 1.2 BLEU improvement for the base model. In the case of the English-Chinese task, we obtain 1.6 BLEU improvement for the baseline model. More importantly, the combination of these strategies finally gets an improvement over the best single strategy with roughly 0.5 BLEU score. We will then give a detailed analysis as followings.

4.1 Encoder v.s. Decoder
As shown in Table 2, pre-trained language model representations are most effective when supervised on the encoder part but less effective on the decoder part. As BERT contains bidirectional information, pre-training decoder may lead to inconsistencies between the training and the inference. The GPT-2 Transformer uses constrained self-attention where every token can only attend to context to its left, thus it is natural to introduce GPT-2 to the NMT decoder. While there are still no more significant gains obtained in our experiments. One possible reason is that the decoder is not a typical language model, which contains the information from source attention. We will leave this issue in the future study.

4.2 BERT v.s. GPT-2
We compare BERT with GPT-2 (Radford et al., 2019, 2018) on WMT 2014 English-German corpus. As shown in Table 2, BERT added encoder works better than GPT-2. The experiments suggest that bidirectional information plays an important role in the encoder of NMT models. While for the decoder part, GPT-2 is a more priority choice. In the following part, we choose BERT as the pre-trained LM and apply only for the encoder part.

Towards Making Most of BERT for NMT, [Yang et al AAAI 2020]
• Cross attention plays a crucial role in NMT
• Pre-trained language models, such as BERT and GPT, have none
• This mismatch between the generation models and conditional generation models makes the pre-trained model usage for translation decoder pretty tricky
PART2: Monolingual Pre-training for NMT

• The Bronze Age
  – NMT initialized with word2vec [ACL 2017, NAACL 2018, AI 2020]
  – NMT initialized with language model [EMNLP 2017]

• BERT fusion
  – BERT Incorporating methods [ICLR 2020, AAAI 2020a]
  – BERT Tuning methods [AAAI 2020b]

• Unified sequence to sequence pre-training
  – MASS: Masked Sequence-to-Sequence Pre-training [ICML 2019]
  – BART: Denoising Sequence-to-Sequence Pre-training [ACL 2020]
MASS: Pre-train for Sequence to Sequence Generation

• MASS is carefully designed to jointly pre-train the encoder and decoder

• Mask k consecutive tokens (segment)
  – Force the decoder to attend on the source representations, i.e., encoder-decoder attention
  – Develop the decoder with the ability of language modeling
MASS vs. BERT/GPT

MASS: Pre-train for Sequence to Sequence Generation, [Song et al ICML 2019]
Unsupervised NMT

XLM: Cross-lingual language model pretraining, [CoRR 2019]

MASS: Pre-train for Sequence to Sequence Generation, [Song et al ICML 2019]
Low-resource NMT

MASS: Pre-train for Sequence to Sequence Generation, [Song et al ICML 2019]
Summary

• Advantages
  – Unified sequence-to-sequence pretraining which jointly pretrains encoder, decoder and cross attention
  – Achieves improvements on zero-shot / unsupervised NMT

• Limitations
  – No experiments on rich resource NMT
  – Pretraining objective inconsistent with NMT, e.g. monolingual v.s. multilingual

MASS: Pre-train for Sequence to Sequence Generation, [Song et al ICML 2019]
A schema comparison with BERT, GPT and BART.

- Standard sequence-to-sequence Transformer architecture
- Trained by corrupting documents and then optimizing a reconstruction loss
- Allows to apply any type of document corruption.
Noising the input

- **Token masking**: Random tokens are sampled and replaced with [MASK].
- **Token deletion**: Random tokens are deleted from the input.
- **Text infilling**: A number of span are sampled. Each span is replaced with [MASK]. 0-length span corresponding the insertion of [MASK].
- **Sentence permutation**: Sentences are shuffled with random order.
- **Document Rotation**: A token is chosen uniformly at random, and the document is rotated so that it begins with that token.
Noising the input

- Token masking: Random tokens are sampled and replaced with [MASK].
- Token deletion: Random tokens are deleted from the input.
- Text infilling: A number of span are sampled. Each span is replaced with [MASK]. 0-length span corresponding the insertion of [MASK].
- Sentence permutation: Sentences are shuffled with random order.
- Document Rotation: A token is chosen uniformly at random, and the document is rotated so that it begins with that token.

BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension, [Lewis et al ACL 2020]
Noising the input

- **Token masking:** Random tokens are sampled and replaced with [MASK].
- **Token deletion:** Random tokens are deleted from the input.
- **Text infilling:** A number of span are sampled. Each span is replaced with [MASK]. 0-length span corresponding the insertion of [MASK].
- **Sentence permutation:** Sentences are shuffled with random order.
- **Document Rotation:** A token is chosen uniformly at random, and the document is rotated so that it begins with that token.
Noising the input

- Token masking: Random tokens are sampled and replaced with [MASK].
- Token deletion: Random tokens are deleted from the input.
- Text infilling: A number of span are sampled. Each span is replaced with [MASK]. 0-length span corresponding the insertion of [MASK].
- Sentence permutation: Sentences are shuffled with random order.
- Document Rotation: A token is chosen uniformly at random, and the document is rotated so that it begins with that token.

BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension, [Lewis et al ACL 2020]
Noising the input

• Token masking: Random tokens are sampled and replaced with [MASK].
• Token deletion: Random tokens are deleted from the input.
• Text infilling: A number of span are sampled. Each span is replaced with [MASK]. 0-length span corresponding the insertion of [MASK].
• Sentence permutation: Sentences are shuffled with random order.
• Document Rotation: A token is chosen uniformly at random, and the document is rotated so that it begins with that token.

BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension, [Lewis et al ACL 2020]
Fine-Tune on Neural Machine Translation

- Replace BART’s encoder embedding layer with a new randomly initialized encoder
- The new encoder uses a separate vocabulary from the original BART mode
- First, freeze BART parameters and only update the randomly initialized source encoder. Then, jointly tuning with a few steps.
Results on IWSLT 2016 En->Ro augmented with back-translation data

- 6 layer of additional transformer encoder to encoding Romania representation.
- *MASS reports unsupervised results
PART III: Multilingual Pre-training for NMT
PART 3: Multilingual Pre-training for NMT

- Multilingual fused pre-training
  - Cross-lingual Language Model Pre-training [NeurIPS, 2019]
  - Alternating Language Modeling Pre-training [AAAI, 2020]
  - XLM-T: Cross-lingual Transformer Encoders

- Multilingual sequence to sequence pre-training
  - mBART [TACL, 2020]
  - CSP [EMNLP, 2020]
  - mRASP & mRASP2 [EMNLP, 2020] [ACL, 2021]
  - LaSS: Learning language-specific sub-network via pre-training & fine-tuning [ACL, 2021]
Multi-lingual Pre-training for NMT

- Data scarcity for low/zero resource languages.
- **Transfer knowledge** between languages.
Learning cross-lingual representation
Similar to BERT, but in many languages…
Multilingual representations emerge from a single model trained on many languages.

Multilingual Masked language modeling pretraining

Cross-lingual Language Model Pre-training, [Conneau et al NeurIPS 2019]
MLM is unsupervised, but TLM leverages parallel data… Encourage the model to learn cross-lingual context when predicting

Translation language modeling (TLM) pretraining

Cross-lingual Language Model Pre-training, [Conneau et al NeurIPS 2019]
Initialization is key in unsupervised MT to bootstrap the iterative BT process.

Embedding layer initialization is essential for neural unsupervised MT (*).

Full Transformer model initialization significantly improves performance (+7 BLEU).

- Embeddings pretrained: 27.3 BLEU
- Full model pretrained (CLM): 30.5 BLEU
- Full model pretrained (MLM): 34.3 BLEU
- Supervised 2016 SOTA (Edinburgh): 36.2 BLEU

Cross-lingual Language Model Pre-training, [Conneau et al NeurIPS 2019]
Results on supervised machine translation

- Pre-training is important for translation
  - Pre-training both encoder and decoder improves
  - MLM is better than CLM
  - Back translation + Pre-training achieve the best
Ablation study

• Adding more languages improves performance on low-resource languages due to positive knowledge transfer

• Sampling batches more often in some languages improves performance in these languages but decrease performance in other languages (capacity allocation problem)

Cross-lingual Language Model Pre-training, [Conneau et al NeurIPS 2019]
Summary

- Cross-lingual language model pre-training is very effective for NMT
- Pre-training reduces the gap between unsupervised and supervised MT
- Encourage knowledge transfer across languages is promising
Alternating Language Modeling for Cross-Lingual Pre-Training

- ALM extend TLM in a sentence, which alternately predicts words of different languages
- ALM can capture the rich cross-lingual context of words and phrases
Overview of ALM pre-training

Sample 1

Source
global monitor and warning satellite system

Target
satellite system

Sample 2

Source
global monitor and warning satellite system

Target
satellite system

Sample n-1

Source
global monitor and warning satellite system

Target
satellite system

Sample n

Source
global monitor and warning satellite system

Target
satellite system
Training details

• Dataset
  – Original parallel data to generate 20 times code-switched sentences
  – Separately obtain the alternating language sentences of source language and target language, which are 40 times than original data
  – Totally, 1.5 billion code-switched sentences are used for pre-training

• Model
  – Transformer big
  – Reload the parameters of ALT for both encoder and decoder. The cross-lingual attention parameters are randomly initialized.
• mBERT: extends the BERT model to different languages
• XLM: the most related work. The results are implemented with released code.
• Mass: set the fragment length k as 50% of the total number of masked tokens in the sentence.

---

**Results**

<table>
<thead>
<tr>
<th>En → De</th>
<th>BLEU(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transformer (Vaswani et al. 2017)</td>
<td>28.40</td>
</tr>
<tr>
<td>ConvS2S (Gehring et al. 2017)</td>
<td>25.16</td>
</tr>
<tr>
<td>Weighted Transformer (Ahmed, Keskar, and Socher 2017)</td>
<td>28.90</td>
</tr>
<tr>
<td>Layer-wise Transformer (He et al. 2018)</td>
<td>29.01</td>
</tr>
<tr>
<td>RNMT+ (Chen et al. 2018)</td>
<td>28.50</td>
</tr>
<tr>
<td>mBERT (Devlin et al. 2019)</td>
<td>28.64</td>
</tr>
<tr>
<td>MASS (Song et al. 2019)</td>
<td>28.92</td>
</tr>
<tr>
<td>XLM (Lample and Conneau 2019)</td>
<td>28.88</td>
</tr>
<tr>
<td><strong>ALM (this work)</strong></td>
<td><strong>29.22</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>De → En</th>
<th>BLEU(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transformer (Vaswani et al. 2017)</td>
<td>34.49</td>
</tr>
<tr>
<td>LightConv (Wu et al. 2019)</td>
<td>34.80</td>
</tr>
<tr>
<td>DynamicConv (Wu et al. 2019)</td>
<td>35.20</td>
</tr>
<tr>
<td>Advsoft (Wang, Gong, and Liu 2019)</td>
<td>35.18</td>
</tr>
<tr>
<td>Layer-wise Transformer (He et al. 2018)</td>
<td>35.07</td>
</tr>
<tr>
<td>mBERT (Devlin et al. 2019)</td>
<td>34.82</td>
</tr>
<tr>
<td>MASS (Song et al. 2019)</td>
<td>35.14</td>
</tr>
<tr>
<td>XLM (Lample and Conneau 2019)</td>
<td>35.22</td>
</tr>
<tr>
<td><strong>ALM (this work)</strong></td>
<td><strong>35.53</strong></td>
</tr>
</tbody>
</table>

---

Alternating Language Modeling for Cross-Lingual Pre-Training [Yang et al AAAI 2020]
• Randomly shuffle the full parallel training set in the task of IWSLT14 German- to-English translation dataset. Then, extract the random K% samples as the fine-tuned parallel data

• Not surprise, the improvements of ALM is larger for low resource NMT
MASS is carefully designed to pre-train the encoder and decoder, unlike BERT that pre-trains only the encoder or the decoder, is an effective transfer learning method that can be applied to other tasks. Universal Language Representation (ULR) (Peters et al. 2018) is proposed as a kind of deep contextualized word representation that is pre-trained in the large scale corpus and can be transferred to other tasks of natural language processing. ELMo (Peters et al. 2018) is an effective transfer learning method that can be applied to other tasks.

ALM can outperform the previous pre-training methods on cross-lingual data. More recently, XLNet (Yang et al. 2019b) proposes a generalized auto-aggressive pre-training method that enables learning bidirectional contexts by maximizing the expected likelihood over all permutations of the factorization and can force the decoder to rely more on the source representation other than the previous tokens in the target side. MASS can convert the code-switched sentences being used for low-resource data.

This work is supported by the National Natural Science Foundation of China (Grand Nos. U1636211, 61672081, 61370126), Beijing Advanced Innovation Center for Imaging Technology (No.BAICIT-2016001) and the Fund of the State Key Laboratory of Software Development Environment (No.SKLSDE-2019ZX-17).
• Initialize MT encoder and decoder with pre-trained cross-lingual encoders
• Fine-tune the model on multilingual parallel data
### Table 1: X $\rightarrow$ En test BLEU for bilingual, many-to-one, and many-to-many models on WMT-10. On the top are the models trained with original parallel data, while the bottom are combined with back-translation. The languages are ordered from high-resource (left) to low-resource (right).

$$
\begin{array}{c|cccccccccc|c}
X \rightarrow En & Fr & Cs & De & Fi & Lv & Et & Ro & Hi & Tr & Gu & Avg \\
\hline
\text{Train on Original Parallel Data (Bitext)} & & & & & & & & & & & \\
\hline
\text{Bilingual NMT} & 36.2 & 28.5 & 40.2 & 19.2 & 17.5 & 19.7 & 29.8 & 14.1 & 15.1 & 9.3 & 23.0 \\
\text{Many-to-One} & 34.8 & 29.0 & 40.1 & 21.2 & 20.4 & 26.2 & 34.8 & 22.8 & 23.8 & 19.2 & 27.2 \\
\text{XLM-T} & 35.9 & 30.5 & 41.6 & 22.5 & 21.4 & 28.4 & 36.6 & 24.6 & 25.6 & 20.4 & 28.8 \\
\hline
\text{Many-to-Many} & 35.9 & 29.2 & 40.0 & 21.1 & 20.4 & 26.3 & 35.5 & 23.6 & 24.3 & 20.6 & 27.7 \\
\text{XLM-T} & 36.0 & 30.0 & 40.8 & 22.1 & 21.5 & 27.8 & 36.5 & 25.3 & 25.0 & 20.6 & 28.5 \\
\hline
\text{Train on Original Parallel Data and Back-Translation Data (Bitext+BT)} & & & & & & & & & & & \\
\hline
\text{(Wang et al., 2020)} & & & & & & & & & & & \\
\text{Many-to-One} & 35.3 & 31.9 & 45.4 & 23.8 & 22.4 & 30.5 & 39.1 & 28.7 & 27.6 & 23.5 & 30.8 \\
\text{XLM-T} & 36.0 & 33.1 & 44.8 & 25.4 & 23.9 & 32.7 & 39.8 & 30.1 & 28.8 & 23.6 & 31.8 \\
\hline
\text{(Wang et al., 2020)} & & & & & & & & & & & \\
\text{Many-to-Many} & 35.3 & 31.2 & 43.7 & 23.1 & 21.5 & 29.5 & 38.1 & 27.5 & 26.2 & 23.4 & 30.0 \\
\text{XLM-T} & 36.1 & 32.6 & 44.3 & 25.4 & 23.8 & 32.0 & 40.3 & 29.5 & 28.7 & 24.2 & 31.7 \\
\end{array}
$$

- The multilingual models achieve much better performance on the low-resource languages and are worse on the high-resource languages
- XLM-T achieves significant improvements over the multilingual baseline across all 10 languages
- In the back-translation setting, XLM-T can further improve this strong baseline
XLM-T: Scaling up Multilingual Machine Translation with Pretrained Cross-lingual Transformer Encoders

<table>
<thead>
<tr>
<th>X → En</th>
<th>Fr</th>
<th>Cs</th>
<th>De</th>
<th>Fi</th>
<th>Lv</th>
<th>Et</th>
<th>Ro</th>
<th>Hi</th>
<th>Tr</th>
<th>Gu</th>
<th>Avg</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Train on Original Parallel Data (Bitext)</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bilingual NMT</td>
<td>36.2</td>
<td>28.5</td>
<td>40.2</td>
<td>19.2</td>
<td>17.5</td>
<td>19.7</td>
<td>29.8</td>
<td>14.1</td>
<td>15.1</td>
<td>9.3</td>
<td>23.0</td>
</tr>
<tr>
<td>Many-to-One</td>
<td>34.8</td>
<td>29.0</td>
<td>40.1</td>
<td>21.2</td>
<td>20.4</td>
<td>26.2</td>
<td>34.8</td>
<td>22.8</td>
<td>23.8</td>
<td>19.2</td>
<td>27.2</td>
</tr>
<tr>
<td>XLM-T</td>
<td>35.9</td>
<td>30.5</td>
<td>41.6</td>
<td>22.5</td>
<td>21.4</td>
<td>28.4</td>
<td>36.6</td>
<td>24.6</td>
<td>25.6</td>
<td>20.4</td>
<td><strong>28.8</strong></td>
</tr>
<tr>
<td>Many-to-Many</td>
<td>35.9</td>
<td>29.2</td>
<td>40.0</td>
<td>21.1</td>
<td>20.4</td>
<td>26.3</td>
<td>35.5</td>
<td>24.3</td>
<td>25.6</td>
<td>20.4</td>
<td><strong>27.7</strong></td>
</tr>
<tr>
<td>XLM-T</td>
<td>35.5</td>
<td>30.0</td>
<td>40.8</td>
<td>22.1</td>
<td>21.5</td>
<td>27.8</td>
<td>36.5</td>
<td>25.3</td>
<td>25.0</td>
<td>20.6</td>
<td><strong>28.5</strong></td>
</tr>
<tr>
<td><strong>Train on Original Parallel Data and Back-Translation Data (Bitext+BT)</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(Wang et al., 2020) Many-to-One</td>
<td>35.3</td>
<td>31.9</td>
<td>45.4</td>
<td>23.8</td>
<td>22.4</td>
<td>30.5</td>
<td>39.1</td>
<td>28.7</td>
<td>27.6</td>
<td>23.5</td>
<td>30.8</td>
</tr>
<tr>
<td>XLM-T</td>
<td>36.0</td>
<td>33.1</td>
<td>44.8</td>
<td>25.4</td>
<td>23.9</td>
<td>32.7</td>
<td>39.8</td>
<td>30.1</td>
<td>28.8</td>
<td><strong>31.8</strong></td>
<td></td>
</tr>
<tr>
<td>(Wang et al., 2020) Many-to-Many</td>
<td>35.3</td>
<td>31.2</td>
<td>43.7</td>
<td>23.1</td>
<td>21.5</td>
<td>29.5</td>
<td>38.1</td>
<td>27.5</td>
<td>26.2</td>
<td>23.4</td>
<td>30.0</td>
</tr>
<tr>
<td>XLM-T</td>
<td>36.1</td>
<td>32.6</td>
<td>44.3</td>
<td>25.4</td>
<td>23.8</td>
<td>32.0</td>
<td>40.3</td>
<td>29.5</td>
<td>28.7</td>
<td>24.2</td>
<td><strong>31.7</strong></td>
</tr>
</tbody>
</table>

- The multilingual models achieve much better performance on the low-resource languages and are worse on the high-resource languages
- XLM-T achieves significant improvements over the multilingual baseline across all 10 languages
- In the back-translation setting, XLM-T can further improve this strong baseline
• Generally, the improvements are smaller than $X \rightarrow En$
• The multilingual part of $En \rightarrow X$ is at the decoder side, which XLM-R is not an expert in.
PART 3: Multilingual Pre-training for NMT

- Multilingual fused pre-training
  - Cross-lingual Language Model Pre-training [NeurIPS, 2019]
  - Alternating Language Modeling Pre-training [AAAI, 2020]
  - XLM-T: Cross-lingual Transformer Encoders

- Multilingual sequence to sequence pre-training
  - mBART [TACL, 2020]
  - CSP [EMNLP, 2020]
  - mRASP & mRASP2 [EMNLP, 2020] [ACL, 2021]
  - LaSS: Learning language-specific sub-network via pre-training & fine-tuning [ACL, 2021]
mBART: Multilingual Denoising Pre-training for Neural Machine Translation

- Multilingual denoising **pre-training** (25 languages)
  - Sentence permutation
  - Word-span masking
- **Fine-tuning** on MT with special language id
Our models follow the BART (Lewis et al., 2019) approach for pre-training on multilingual data.

### 2.2 Model: mBART

We fine-tune on additional languages. We do not apply additional preprocessing, such as true-case, punctuation, or special tokens. Instead, we use the smoothing parameter \( \lambda_i = \frac{1}{p_i} \cdot \frac{p_i^\alpha}{\sum_i p_i^\alpha} \), where \( p_i \) is the percentage of each language in the Common Crawl (CC) corpus. Following Lample and Conneau (2019), we re-tokenize data to be used for pre-training, this tokenization supports fine-tuning on additional languages.

We use a large-scale common crawl (CC) corpus from each language (§2.3). We tokenize with Sentence Piece which includes 25,000 subwords. While not all of these languages are used for pre-training, this tokenization supports fine-tuning on additional languages.

We use the smoothing parameter \( \lambda_i = \frac{1}{p_i} \cdot \frac{p_i^\alpha}{\sum_i p_i^\alpha} \), where \( p_i \) is the percentage of each language in the CC corpus.

We also show that mBART enables new types of transfer across language pairs. For example, in experiments in the later sections involve fine-tuning on bi-text in one language pair (e.g., German-English), with no further training. We also show that languages not learned on the full CC data that includes multiple languages – CC25 – extracted from the Common Crawl (CC) benefit from mBART, strongly suggesting that the initialization is at least partially language universal. Finally, we present a detailed analysis of which factors contribute the most to effective pre-training, including the number of monolingual documents in language names with their ISO codes for simplicity. (*).

### Table 1: Languages, Tokens/M, Size/GB

<table>
<thead>
<tr>
<th>Code</th>
<th>Language</th>
<th>Tokens/M</th>
<th>Size/GB</th>
</tr>
</thead>
<tbody>
<tr>
<td>En</td>
<td>English</td>
<td>55608</td>
<td>300.8</td>
</tr>
<tr>
<td>Ru</td>
<td>Russian</td>
<td>23408</td>
<td>278.0</td>
</tr>
<tr>
<td>Vi</td>
<td>Vietnamese</td>
<td>24757</td>
<td>137.3</td>
</tr>
<tr>
<td>Ja</td>
<td>Japanese</td>
<td>530 (*)</td>
<td>69.3</td>
</tr>
<tr>
<td>De</td>
<td>German</td>
<td>10297</td>
<td>66.6</td>
</tr>
<tr>
<td>Ro</td>
<td>Romanian</td>
<td>10354</td>
<td>61.4</td>
</tr>
<tr>
<td>Fr</td>
<td>French</td>
<td>9780</td>
<td>56.8</td>
</tr>
<tr>
<td>Fi</td>
<td>Finnish</td>
<td>6730</td>
<td>54.3</td>
</tr>
<tr>
<td>Ko</td>
<td>Korean</td>
<td>5644</td>
<td>54.2</td>
</tr>
<tr>
<td>Es</td>
<td>Spanish</td>
<td>9374</td>
<td>53.3</td>
</tr>
<tr>
<td>Zh</td>
<td>Chinese (Sim)</td>
<td>259 (*)</td>
<td>46.9</td>
</tr>
<tr>
<td>It</td>
<td>Italian</td>
<td>4983</td>
<td>30.2</td>
</tr>
<tr>
<td>Nl</td>
<td>Dutch</td>
<td>5025</td>
<td>29.3</td>
</tr>
<tr>
<td>Ar</td>
<td>Arabic</td>
<td>2869</td>
<td>28.0</td>
</tr>
<tr>
<td>Tr</td>
<td>Turkish</td>
<td>2736</td>
<td>20.9</td>
</tr>
<tr>
<td>Hi</td>
<td>Hindi</td>
<td>1715</td>
<td>20.2</td>
</tr>
<tr>
<td>Cs</td>
<td>Czech</td>
<td>2498</td>
<td>16.3</td>
</tr>
<tr>
<td>Lt</td>
<td>Lithuanian</td>
<td>1835</td>
<td>13.7</td>
</tr>
<tr>
<td>Lv</td>
<td>Latvian</td>
<td>1198</td>
<td>8.8</td>
</tr>
<tr>
<td>Kk</td>
<td>Kazakh</td>
<td>476</td>
<td>6.4</td>
</tr>
<tr>
<td>Et</td>
<td>Estonian</td>
<td>843</td>
<td>6.1</td>
</tr>
<tr>
<td>Ne</td>
<td>Nepali</td>
<td>237</td>
<td>3.8</td>
</tr>
<tr>
<td>Si</td>
<td>Sinhala</td>
<td>243</td>
<td>3.6</td>
</tr>
<tr>
<td>Gu</td>
<td>Gujarati</td>
<td>140</td>
<td>1.9</td>
</tr>
<tr>
<td>My</td>
<td>Burmese</td>
<td>56</td>
<td>1.6</td>
</tr>
</tbody>
</table>
mBART: Low-medium translation results

Table 2: Low/Medium Resource Machine Translation
Pre-training consistently improves over a randomly initialized baseline, with particularly large gains on low resource language pairs (e.g. Vi-En).

<table>
<thead>
<tr>
<th>Languages</th>
<th>Data Source</th>
<th>Size</th>
<th>Direction</th>
<th>Random</th>
<th>mBART25</th>
</tr>
</thead>
<tbody>
<tr>
<td>En-Gu</td>
<td>WMT19</td>
<td>10K</td>
<td>← →</td>
<td>0.0</td>
<td>0.3</td>
</tr>
<tr>
<td>En-Kk</td>
<td>WMT19</td>
<td>91K</td>
<td>← →</td>
<td>0.8</td>
<td>0.1</td>
</tr>
<tr>
<td>En-Vi</td>
<td>IWSLT15</td>
<td>133K</td>
<td>← →</td>
<td>0.2</td>
<td>7.4</td>
</tr>
<tr>
<td>En-Tr</td>
<td>WMT17</td>
<td>207K</td>
<td>← →</td>
<td>23.6</td>
<td>36.1</td>
</tr>
<tr>
<td>En-Ja</td>
<td>IWSLT17</td>
<td>223K</td>
<td>← →</td>
<td>24.8</td>
<td>35.4</td>
</tr>
<tr>
<td>En-Ko</td>
<td>IWSLT17</td>
<td>230K</td>
<td>← →</td>
<td>12.2</td>
<td>22.5</td>
</tr>
</tbody>
</table>

Table 3: High Resource Machine Translation
where all the datasets are from their latest WMT competitions. We only evaluate our models on En-X translation.

<table>
<thead>
<tr>
<th>Languages</th>
<th>Data Source</th>
<th>Size</th>
<th>Direction</th>
<th>Random</th>
<th>mBART25</th>
</tr>
</thead>
<tbody>
<tr>
<td>En-Nl</td>
<td>IWSLT17</td>
<td>237K</td>
<td>← →</td>
<td>34.6</td>
<td>43.3</td>
</tr>
<tr>
<td>En-Ar</td>
<td>IWSLT17</td>
<td>250K</td>
<td>← →</td>
<td>29.3</td>
<td>34.8</td>
</tr>
<tr>
<td>En-It</td>
<td>IWSLT17</td>
<td>250K</td>
<td>← →</td>
<td>27.5</td>
<td>37.6</td>
</tr>
<tr>
<td>En-My</td>
<td>WAT19</td>
<td>259K</td>
<td>← →</td>
<td>16.9</td>
<td>21.6</td>
</tr>
<tr>
<td>En-Ne</td>
<td>FLoRes</td>
<td>564K</td>
<td>← →</td>
<td>31.7</td>
<td>39.8</td>
</tr>
<tr>
<td>En-Ro</td>
<td>WMT16</td>
<td>608K</td>
<td>← →</td>
<td>28.0</td>
<td>34.0</td>
</tr>
</tbody>
</table>

3.1 Experimental Settings

Datasets
We gather 24 pairs of publicly available parallel corpora that cover all the languages in CC25 (Table 1). Most pairs are from previous WMT (Gu, Kk, Tr, Ro, Et, Lt, Fi, Lv, Cs, Es, Zh, De, Ru, Fr) and IWSLT (Vi, Ja, Ko, Nl, Ar, It) competitions. We also use FLoRes pairs (Guzmán et al., 2019, En-Ne and En-Si), En-Hi from IITB (Kunchukuttan et al., 2017), and En-My from WAT19 (Ding et al., 2018, 2019).

We divide the datasets into three categories – low resource (<1M sentence pairs), medium resource (>1M and <10M), and high resource (>10M).

Fine-tuning & Decoding
We fine-tune our multilingual pre-trained models on a single pair of bi-text data, feeding the source language into the encoder and decoding the target language. As shown in Figure 1, we load the pre-trained weights and train the MT model on bi-texts with teacher forcing. For all directions, we train with 0.3 dropout, 0.2 label smoothing, 2500 warm-up steps, 3e-5 maximum learning rate. We use a maximum of 40K training updates for all low and medium resource pairs and 100K for high resource pairs. The final models are selected based on validation likelihood. For decoding, we use beam-search with beam size 5 for all directions. The final results are reported in BLEU (Papineni et al., 2002) with language-specific settings, see appendix A.

3.2 Main Results
As shown in Table 2, initializing with the pre-trained mBART25 weights shows gains on all the low and medium resource pairs when compared with randomly initialized baselines. We observe gains of 12+ BLEU on low resource pairs such as En-Vi, En-Tr, and noisily aligned pairs like En-Hi.

Low resource: more than 6 BLEU. But fails in extremely low-resource setting
### mBART: Low-medium translation results

#### Table 2: Low/Medium Resource Machine Translation

<table>
<thead>
<tr>
<th>Languages</th>
<th>En-Gu</th>
<th>En-Kk</th>
<th>En-Vi</th>
<th>En-Tr</th>
<th>En-Ja</th>
<th>En-Ko</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Source</td>
<td>WMT19</td>
<td>WMT19</td>
<td>IWSLT15</td>
<td>WMT17</td>
<td>IWSLT17</td>
<td>IWSLT17</td>
</tr>
<tr>
<td>Size</td>
<td>10K</td>
<td>91K</td>
<td>133K</td>
<td>207K</td>
<td>223K</td>
<td>230K</td>
</tr>
<tr>
<td>Direction</td>
<td>← →</td>
<td>← →</td>
<td>← →</td>
<td>← →</td>
<td>← →</td>
<td>← →</td>
</tr>
<tr>
<td>Random</td>
<td>0.0</td>
<td>0.8</td>
<td>23.6</td>
<td>12.2</td>
<td>10.4</td>
<td>15.3</td>
</tr>
<tr>
<td>mBART25</td>
<td>0.3</td>
<td>7.4</td>
<td>36.1</td>
<td>22.5</td>
<td>19.4</td>
<td>24.6</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Languages</th>
<th>En-Nl</th>
<th>En-Ar</th>
<th>En-It</th>
<th>En-My</th>
<th>En-Ne</th>
<th>En-Ro</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Source</td>
<td>IWSLT17</td>
<td>IWSLT17</td>
<td>IWSLT17</td>
<td>WAT19</td>
<td>FLoRes</td>
<td>WMT16</td>
</tr>
<tr>
<td>Size</td>
<td>237K</td>
<td>250K</td>
<td>250K</td>
<td>259K</td>
<td>564K</td>
<td>608K</td>
</tr>
<tr>
<td>Direction</td>
<td>← →</td>
<td>← →</td>
<td>← →</td>
<td>← →</td>
<td>← →</td>
<td>← →</td>
</tr>
<tr>
<td>Random</td>
<td>34.6</td>
<td>27.5</td>
<td>31.7</td>
<td>23.3</td>
<td>7.6</td>
<td>34.0</td>
</tr>
<tr>
<td>mBART25</td>
<td>43.3</td>
<td>37.6</td>
<td>39.8</td>
<td>28.3</td>
<td>14.5</td>
<td>37.8</td>
</tr>
</tbody>
</table>

#### Table 3: High Resource Machine Translation

<table>
<thead>
<tr>
<th>Languages</th>
<th>En-Si</th>
<th>En-Hi</th>
<th>En-Et</th>
<th>En-Lt</th>
<th>En-Fi</th>
<th>En-Lv</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Source</td>
<td>FLoRes</td>
<td>ITTB</td>
<td>WMT18</td>
<td>WMT19</td>
<td>WMT17</td>
<td>WMT17</td>
</tr>
<tr>
<td>Size</td>
<td>647K</td>
<td>1.56M</td>
<td>1.94M</td>
<td>2.11M</td>
<td>2.66M</td>
<td>4.50M</td>
</tr>
<tr>
<td>Direction</td>
<td>← →</td>
<td>← →</td>
<td>← →</td>
<td>← →</td>
<td>← →</td>
<td>← →</td>
</tr>
<tr>
<td>Random</td>
<td>7.2</td>
<td>10.9</td>
<td>22.6</td>
<td>18.1</td>
<td>18.1</td>
<td>15.6</td>
</tr>
<tr>
<td>mBART25</td>
<td>13.7</td>
<td>23.5</td>
<td>27.8</td>
<td>22.4</td>
<td>28.5</td>
<td>19.3</td>
</tr>
</tbody>
</table>

#### Low resource: more than 6 BLEU. But fails in extremely low-resource setting

#### Medium resource: more than 3 BLEU
mBART: Rich-resource translation

Table 2: Low/Medium Resource Machine Translation

Pre-training consistently improves over a randomly initialized baseline, with particularly large gains on low resource language pairs (e.g. Vi-En).

Table 3: High Resource Machine Translation

where all the datasets are from their latest WMT competitions. We only evaluate our models on En-X translation.

3 Sentence-level Machine Translation

This section shows that mBART pre-training provides consistent performance gains in low to medium resource sentence-level MT settings, including bi-text only and with back translation, and outperforms other existing pre-training schemes (§3.2). We also present a detailed analysis to understand better which factors contribute the most to these gains (§3.3), and show that pre-training can even improve performance for languages not present in the pre-training data at all (§3.4).

3.1 Experimental Settings

Datasets

We gather 24 pairs of publicly available parallel corpora that cover all the languages in CC25 (Table 1). Most pairs are from previous WMT (Gu, Kk, Tr, Ro, Et, Lt, Fi, Lv, Cs, Es, Zh, De, Ru, Fr $^\text{En}$) and IWSLT (Vi, Ja, Ko, Nl, Ar, It $^\text{En}$) competitions. We also use FLoRes pairs (Guzmán et al., 2019, En-Ne and En-Si), En-Hi from IITB (Kunchukuttan et al., 2017), and En-My from WAT19 (Ding et al., 2018, 2019).

We divide the datasets into three categories – low resource (<1M sentence pairs), medium resource (>1M and <10M), and high resource (>10M).

Fine-tuning & Decoding

We fine-tune our multilingual pre-trained models on a single pair of bi-text data, feeding the source language into the encoder and decoding the target language. As shown in Figure 1, we load the pre-trained weights and train the MT model on bi-texts with teacher forcing. For all directions, we train with 0.3 dropout, 0.2 label smoothing, 2500 warm-up steps, 3e-5 maximum learning rate. We use a maximum of 40K training updates for all low and medium resource pairs and 100K for high resource pairs. The final models are selected based on validation likelihood. For decoding, we use beam-search with beam size 5 for all directions. The final results are reported in BLEU (Papineni et al., 2002) with language-specific settings, see appendix A.

3.2 Main Results

As shown in Table 2, initializing with the pre-trained mBART25 weights shows gains on all the low and medium resource pairs when compared with randomly initialized baselines. We observe gains of 12+ BLEU on low resource pairs such as En-Vi, En-Tr, and noisily aligned pairs like En-Hi. Fine-tuning fails in extremely low-resource setting such as En-Gu, which only have roughly 10k examples.

Pre-training slightly hurts performance when >25M parallel sentence are available.

When a significant amount of bi-text data is given, supervised training are supposed to wash out the pre-trained weights completely.
mBART: Pre-training complementary to BT

- Test on low resource FLoRes dataset [Guzmán et al., 2019]
- Use the same monolingual data to generate BT data
- Initializing the model with mBART25 pre-trained parameters improves BLEU scores at each iteration of back translation, resulting in new state-of-the-art results in all four translation directions
Is pre-training on multilingual better than on single language?

<table>
<thead>
<tr>
<th>Pre-training Model</th>
<th>Data</th>
<th>Fine-tuning En→Ro</th>
<th>Ro→En</th>
<th>+BT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Random</td>
<td>None</td>
<td>34.3</td>
<td>34.0</td>
<td>36.8</td>
</tr>
<tr>
<td>XLM (2019)</td>
<td>En Ro</td>
<td>-</td>
<td>35.6</td>
<td>38.5</td>
</tr>
<tr>
<td>MASS (2019)</td>
<td>En Ro</td>
<td>-</td>
<td>-</td>
<td>39.1</td>
</tr>
<tr>
<td>BART (2019)</td>
<td>En</td>
<td>-</td>
<td>-</td>
<td>38.0</td>
</tr>
<tr>
<td>XLM-R (2019)</td>
<td>CC100</td>
<td>35.6</td>
<td>35.8</td>
<td>-</td>
</tr>
<tr>
<td>BART-En</td>
<td>En</td>
<td>36.0</td>
<td>35.8</td>
<td>37.4</td>
</tr>
<tr>
<td>BART-Ro</td>
<td>Ro</td>
<td>37.6</td>
<td>36.8</td>
<td>38.1</td>
</tr>
<tr>
<td>mBART02</td>
<td>En Ro</td>
<td><strong>38.5</strong></td>
<td><strong>38.5</strong></td>
<td><strong>39.9</strong></td>
</tr>
<tr>
<td>mBART25</td>
<td>CC25</td>
<td>37.7</td>
<td>37.8</td>
<td>38.8</td>
</tr>
</tbody>
</table>

- BART model trained on the same En and Ro data only. Both have improvements over baselines, while worse than mBART results, indicating pre-training in a multilingual setting is essential.
- Combining BT leads to additional gains, resulting in a new state-of-the-art for Ro-En translation.
- mBART02 is better than mBART25. The more seems not the better?
How many languages should you pre-train on?

- Pretraining on more languages helps most when the target language monolingual data is limited.
- When monolingual data is plentiful (De, Ro), pre-training on multiple languages slightly hurts the final results (<1 BLEU).

### Table 5: Pretraining Languages on En-X translation.

<table>
<thead>
<tr>
<th>Languages</th>
<th>De</th>
<th>Ro</th>
<th>It</th>
<th>My</th>
<th>En</th>
</tr>
</thead>
<tbody>
<tr>
<td>Size/GB</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>300.8</td>
</tr>
<tr>
<td>mBART02</td>
<td>31.3</td>
<td>38.5</td>
<td>39.7</td>
<td>36.5</td>
<td></td>
</tr>
<tr>
<td>mBART06</td>
<td>-</td>
<td>38.5</td>
<td>39.3</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>mBART25</td>
<td>30.5</td>
<td>37.7</td>
<td>39.8</td>
<td>36.9</td>
<td></td>
</tr>
</tbody>
</table>

### Figure 2: Comparison with Back-Translation on My-En translation using same monolingual data. We also estimate the computational costs for both pre-training and back-translation based on Nvidia V100 GPUs.

### Figure 4: Fine-tuning curves for En-De along with size of bitext. The x-axis is on a log scale.

### Figure 3: Fine-tuning curves for Ro-En along with Pre-training steps. Both mBART25 and mBART02 outperform the best baseline system after 25K steps.

Multilingual Denoising Pre-training for Neural Machine Translation [Liu et al., TACL 2020]
• Without any pre-training, the model overfits and performs much worse than the baseline.
• After just 25K steps (5% of training), both models outperform the best baseline.
• The models keep improving by over 3 BLEU for the rest of steps and have not fully converged after 500K steps.
• The more the better.

Analysis: Pre-training steps matters

![Fine-tuning curves for Ro-En along with Pre-training steps](image)

![Fine-tuning curves for En-De along with size of bitext](image)

The x-axis is on a log scale.

Table 5:
Pretraining Languages on En-X translation.
The size refers to the size of monolingual data for X. The size of En is shown as reference. All the pretrained models were controlled to see the same number of En-English instances during training.

<table>
<thead>
<tr>
<th>Language</th>
<th>En</th>
<th>My</th>
<th>De</th>
<th>It</th>
</tr>
</thead>
<tbody>
<tr>
<td>Size/GB</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>En-My</td>
<td>300.8</td>
<td>31.3</td>
<td>38.5</td>
<td>39.7</td>
</tr>
<tr>
<td>mBART02</td>
<td>350</td>
<td>34.9</td>
<td>39.2</td>
<td></td>
</tr>
<tr>
<td>mBART06</td>
<td>300</td>
<td>34.9</td>
<td>39.2</td>
<td></td>
</tr>
<tr>
<td>mBART25</td>
<td>300</td>
<td>34.9</td>
<td>39.2</td>
<td></td>
</tr>
</tbody>
</table>

Table 6: Comparison with Back-Translation on My-En translation using same monolingual data. We also estimate the computational costs for both pre-training and back-translation based on Nvidia V100 GPUs.

Figure 2 presents that our pre-trained models can be combined with iterative back-translation (BT) on additional data, however, it is still not a fair comparison. Table 6 shows the results when using same monolingual data where we use 79M En and 29M My sentences following Chen et al. (2019).

With the same amount of monolingual corpus, mBART pre-training achieves the same performance on En!My as BT, while still 3 BLEU worse on My!En. We suspect BT benefits from bigger monolingual data (En). Moreover, combining mBART02 model with BT, we see further gains even with same monolingual data. Besides, we also provide estimated training costs where BT has a longer pipeline involving training a baseline system (5h), translating monolingual data (300h) and formal training (350h). Instead, most of training costs of mBART lies in the pre-training part and can be easily adjusted to be more efficient.

3.4 Generalization to Languages NOT in Pre-training

In this section, we show that mBART can improve performance even with fine tuning for languages that did not appear in the pre-training corpora, suggesting that the pre-training has language universal aspects, especially within the parameters learned at the Transformer layers.

Multilingual Denoising Pre-training for Neural Machine Translation [Liu et al., TACL 2020]
The pre-trained model is able to achieve over 20 BLEU with only 10K training examples, while the baseline system scores 0.

Unsurprisingly, mBART consistently outperforms the baseline models, but the gap reduces with increasing amounts of bi-text, especially after 10M sentence pairs.

Analysis: Perform better on low resource languages

<table>
<thead>
<tr>
<th>Languages</th>
<th>Size/GB</th>
<th>mBART02 BLEU</th>
<th>mBART06 BLEU</th>
<th>mBART25 BLEU</th>
</tr>
</thead>
<tbody>
<tr>
<td>De</td>
<td>66.6</td>
<td>31.3</td>
<td>38.5</td>
<td>30.5</td>
</tr>
<tr>
<td>Ro</td>
<td>61.4</td>
<td>38.5</td>
<td>39.7</td>
<td>37.7</td>
</tr>
<tr>
<td>It</td>
<td>30.2</td>
<td>36.5</td>
<td>39.3</td>
<td>36.9</td>
</tr>
<tr>
<td>My</td>
<td>1.6</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>En</td>
<td>300.8</td>
<td>300.8</td>
<td>34.9</td>
<td>39.2</td>
</tr>
</tbody>
</table>

Table 5: Pretraining Languages on En-X translation. The size refers to the size of monolingual data for X. The size of En is shown as reference. All the pretrained models were controlled to see the same number of En-English instances during training.

Table 6: Comparison with Back-Translation on My-En translation using same monolingual data. We also estimate the computational costs for both pre-training and back-translation based on Nvidia V100 GPUs.

Is pre-training complementary to BT? Figure 2 presents that our pre-trained models can be combined with iterative back-translation (BT) on additional data, however, it is still not a fair comparison. Table 6 shows the results when using back-translation with the same amount of monolingual corpus, mBART pre-training achieves the same performance on En!My as BT, while still 3 BLEU worse on My!En. We suspect BT benefits from bigger monolingual data (En). Moreover, combining mBART02 model with BT, we see further gains even with same monolingual data. Besides, we also provide estimated training costs where BT has a longer pipeline involving training a baseline system (5h), translating monolingual data (300h) and formal training (350h). Instead, most of training costs of mBART lies in the pre-training part and can be easily adjusted to be more efficient.

3.4 Generalization to Languages NOT in Pre-training

In this section, we show that mBART can improve performance even with fine tuning for languages that did not appear in the pre-training corpora, suggesting that the pre-training has language universal aspects, especially within the parameters learned at the Transformer layers.
Analysis: Generalization to unseen languages

<table>
<thead>
<tr>
<th>Monolingual</th>
<th>Nl-En</th>
<th>En-Nl</th>
<th>Ar-En</th>
<th>En-Ar</th>
<th>Ni-De</th>
<th>De-Nl</th>
</tr>
</thead>
<tbody>
<tr>
<td>Random</td>
<td>None</td>
<td>34.6 (-8.7)</td>
<td>29.3 (-5.5)</td>
<td>27.5 (-10.1)</td>
<td>16.9 (-4.7)</td>
<td>21.3 (-6.4)</td>
</tr>
<tr>
<td>mBART02</td>
<td>En Ro</td>
<td>41.4 (-2.9)</td>
<td>34.5 (-0.3)</td>
<td>34.9 (-2.7)</td>
<td>21.2 (-0.4)</td>
<td>26.1 (-1.6)</td>
</tr>
<tr>
<td>mBART06</td>
<td>En Ro Cs It Fr Es</td>
<td>43.1 (-0.2)</td>
<td>34.6 (-0.2)</td>
<td>37.3 (-0.3)</td>
<td>21.1 (-0.5)</td>
<td>26.4 (-1.3)</td>
</tr>
<tr>
<td>mBART25</td>
<td>All</td>
<td>43.3</td>
<td>34.8</td>
<td>37.6</td>
<td>21.6</td>
<td>27.7</td>
</tr>
</tbody>
</table>

Ni-De and Ar are not included in the pre-training corpus

- mBART can improve performance even with fine tuning for languages that did not appear in the pre-training corpora,
- Pre-training has language universal aspects, especially within the parameters learned at the Transformer layers.
- The more pre-trained languages the better
Unsupervised Machine Translation

- Following the same procedure with UNMT, but initialize the translation model with the pre-trained mBART.
- To avoid simply copying the source text, constrain mBART to only generating tokens in target language.
- Achieve very competitive results.

---

**Unsupervised MT via Back-Translation**

- Input
- Monolingual Ne Text
- MLE loss
- Input
- Monolingual En Text
- MLE loss
- Decode
- Generated En Text

UNMT with back translation

---

**Unsupervised MT via Language Transfer**

1. Back-Translation
2. Language Transfer
3. Evaluation Regime

---

**Datasets**

- Both approaches are presented in Figure 5.
- We mask out the output probability of predicting tokens in target language with the pre-trained mBART.
- Avoid it simply copying the source text.
- mBART to only generating tokens in target language.
- However, we do constrain the back-translation (BT).

---

**Learning**

- Following the same procedure described in Table 12.
- We present the direct fine-tuning performance (§3) on un-supervised MT with back translation.

---

**BLEU**

- Table 10:
- ``Model`` | ``Similar Pairs`` | ``Dissimilar Pairs``
- |  | ``En-De`` | ``En-Ro`` | ``En-Ne`` | ``En-Si``
- Random | 21.0 | 17.2 | 19.4 | 21.2 | 0.0 | 0.0 | 0.0 | 0.0
- XLM (2019) | 34.3 | 26.4 | 31.8 | 33.3 | 0.5 | 0.1 | 0.1 | 0.1
- MASS (2019) | 35.2 | 28.3 | 33.1 | 35.2 | - | - | - | -
- mBART | 34.0 | 29.8 | 30.5 | 35.0 | 10.0 | 4.4 | 8.2 | 3.9

---

**Evaluation Regime**

- Table 11: Performance comparison for Unsupervised MT via Language Transfer.
- As illustrated in Figure 5 (b), we take the second case of unsupervised machine translation.
- This work inherits from the recent success brought by self-training.
- The second case of unsupervised machine translation.

---

**Related Work**

- The second case of unsupervised machine translation.
- This work inherits from the recent success brought by self-training.

---

**Multilingual Denoising Pre-training for Neural Machine Translation** [Liu et al., TACL 2020]
CSP: Code-Switching Pre-training for Neural Machine Translation

- Sequence-level pre-training with only monolingual data
- Sub-span of the source sentence is replaced with their lexical translation

The training paradigm follows MASS

Lexical translation is build with only monolingual data.
[Learning bilingual word embeddings with (almost) no bilingual data.]
CSP: Code-Switching Pre-training for Neural Machine Translation

<table>
<thead>
<tr>
<th>System</th>
<th>en-de</th>
<th>de-en</th>
<th>en-fr</th>
<th>fr-en</th>
<th>zh-en</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lample et al. (2018b)</td>
<td>17.16</td>
<td>21.0</td>
<td>25.14</td>
<td>24.18</td>
<td></td>
</tr>
<tr>
<td>Lample and Conneau (2019)</td>
<td>27.0</td>
<td>34.3</td>
<td>33.4</td>
<td>33.3</td>
<td></td>
</tr>
<tr>
<td>Song et al. (2019b)</td>
<td>28.1</td>
<td>35.0</td>
<td>37.5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Lample and Conneau (2019) (our reproduction)</td>
<td>27.3</td>
<td>33.8</td>
<td>32.9</td>
<td>33.5</td>
<td>22.1</td>
</tr>
<tr>
<td>Song et al. (2019b) (our reproduction)</td>
<td>27.9</td>
<td>34.7</td>
<td>37.3</td>
<td>34.1</td>
<td>22.8</td>
</tr>
<tr>
<td>CSP and fine-tuning (ours)</td>
<td>28.7</td>
<td>35.7</td>
<td>37.9</td>
<td>34.5</td>
<td>23.9</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>System</th>
<th>en-de</th>
<th>en-fr</th>
<th>zh-en</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vaswani et al. (2017)</td>
<td>27.3</td>
<td>38.1</td>
<td></td>
</tr>
<tr>
<td>Vaswani et al. (2017) (our reproduction) / + BT</td>
<td>27.0 / 28.6</td>
<td>37.9 / 39.3</td>
<td>42.1 / 43.7</td>
</tr>
<tr>
<td>Lample and Conneau (2019) (our reproduction) / + BT</td>
<td>28.1 / 29.4</td>
<td>38.3 / 39.6</td>
<td>42.0 / 43.7</td>
</tr>
<tr>
<td>Song et al. (2019b) (our reproduction) / + BT</td>
<td>28.4 / 29.6</td>
<td>38.4 / 39.6</td>
<td>42.5 / 44.1</td>
</tr>
<tr>
<td>CSP and fine-tuning (ours) / + BT</td>
<td>28.9 / 30.0</td>
<td>38.8 / 39.9</td>
<td>43.2 / 44.6</td>
</tr>
</tbody>
</table>
mRASP: multilingual Random Aligned Substitution Pre-training

• **mRASP**: multilingual Random Aligned Substitution Pre-training
  - Multilingual Pre-training Approach
  - RAS: specially designed training method to align semantic embeddings

Pre-training Multilingual Neural Machine Translation by Leveraging Alignment Information  [Lin et al., EMNLP 2020]
mRASP: Overview

Pre-training Multilingual Neural Machine Translation by Leveraging Alignment Information [Lin et al., EMNLP 2020]
mRASP: Overview

Pre-training

Encoder

Decoder

Orig

tok

pos

RAS

tok

pos

Fine-tuning

En-Fr
mRASP: RAS method

- Random Aligned Substitution (RAS)
  - Randomly replace a source word to its synonym in different language.
  - Draw the embedding space closer.

\[
\mathcal{L}^{pre} = \sum_{i,j \in \mathcal{E}} \mathbb{E}_{(x^i, x^j) \sim \mathcal{D}_{i,j}} \left[ -\log P_{\theta} \left( x^i \mid C \left( x^j \right) \right) \right]
\]
Training Data for mRASP

- Pre-training Dataset: PC32 (Parallel Corpus 32)
  - 32 English-centric language pairs, resulting in 64 directed translation pairs in total
  - Contains a total size of 110.4M public parallel sentence pairs

![Graph showing the number of En-X sentence pairs for different languages]

Pre-training Multilingual Neural Machine Translation by Leveraging Alignment Information [Lin et al., EMNLP 2020]
mRASP: Fine-tuning Dataset

- Fine-tuning Dataset
- **Indigenous** Corpus: included in pre-training phase
  - Extremely low resource (<100K) (Be, My, etc.)
  - Low resource (>100k and <1M) (He, Tr, etc.)
  - Medium resource (>1M and <10M) (De, Et, etc.)
  - Rich resource (>10M) (Zh, Fr, etc.)
mRASP: Rich resource works

- Rich resource benchmarks can be further improved (En->Fr +1.1BLEU).
mRASP: Low resource works

Pre-training Multilingual Neural Machine Translation by Leveraging Alignment Information [Lin et al., EMNLP 2020]
mRASP: Unseen languages

- mRASP generalizes on all exotic scenarios.

<table>
<thead>
<tr>
<th>Exotic Pair</th>
<th>Fr-Zh(20K)</th>
<th>De-Fr(9M)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Direct</td>
<td>0.7</td>
<td>3</td>
</tr>
<tr>
<td>mRASP</td>
<td>25.8</td>
<td>26.7</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Exotic Pair</th>
<th>Ni-Pt(12K)</th>
<th>Da-El(1.2M)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Direct</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>mRASP</td>
<td>14.1</td>
<td>13.2</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Exotic Source/Target</th>
<th>En-Mr(11K)</th>
<th>En-Gl(1.2M)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Direct</td>
<td>6.4</td>
<td>6.8</td>
</tr>
<tr>
<td>mRASP</td>
<td>22.7</td>
<td>22.9</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Exotic Source/Target</th>
<th>En-Eu(726k)</th>
<th>En-Sl(2M)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Direct</td>
<td>7.1</td>
<td>10.9</td>
</tr>
<tr>
<td>mRASP</td>
<td>19.1</td>
<td>28.4</td>
</tr>
</tbody>
</table>

12k: Direct not work VS mRASP achieves 10+ BLEU!!

Pre-training Multilingual Neural Machine Translation by Leveraging Alignment Information [Lin et al., EMNLP 2020]
mRASP: Compare with other methods

- mRASP outperforms mBART for all but two language pairs.

Pre-training Multilingual Neural Machine Translation by Leveraging Alignment Information [Lin et al., EMNLP 2020]
mRASP: Makes multilingual embeddings more similar

RAS draws the embedding space of languages closer.

Pre-training Multilingual Neural Machine Translation by Leveraging Alignment Information [Lin et al., EMNLP 2020]
mRASP 2: Contrastive Learning for Many-to-many Multilingual Neural Machine Translation

- **Supervised**
  - ✔️
- **Unsupervised**
  - ✔️
- **Zero-shot**
  - ✔️

Enabling unsupervised / zero-shot translation

- **Parallel**
  - ✔️
- **Monolingual**
  - ✔️

Leveraging both parallel & monolingual data

Comparable / better performance on high-resource directions
mRASP2 introduces monolingual data

- Parallel text

- Monolingual text

Contrastive Learning for Many-to-many Multilingual Neural Machine Translation [Pan et al., ACL 2021]
mRASP2 maps different languages in a same space

Contrastive Learning for Many-to-many Multilingual Neural Machine Translation [Pan et al., ACL 2021]
Experiments

Monolingual Corpus mainly contributes to unsupervised translation
Better Semantic Alignment: Sentence Retrieval

15-way parallel test set (Ted-M): 2284 samples

Contrastive Learning and Aligned Augmentation both contribute to the improvement on sentence retrieval
LaSS accommodates one sub-network for each language pair.

- Each language pair has **shared parameters** with some other language pairs and preserves its **language-specific parameters**
- For fine-tuning, only updates the corresponding parameters
Efficacy in alleviating Parameter Interference

– WMT

LaSS obtains consistent gains for both Transformer-base and Transformer-big

Learning Language Specific Sub-network for Multilingual Machine Translation [Lin et al., ACL 2021]
LaSS obtains more gains for rich resource

- WMT

With the dataset scale increasing, the improvement becomes larger, since rich resource language pairs suffer more from parameter interference.
Adaptation to New Language Pairs

• Distribute a new sub-network for new language pair and train the sub-network for fixed steps

![Graph showing BLEU scores over steps for different models and directions. The graph compares Bilingual models with other models like model, LaSS, baseline, and direction. The y-axis represents BLEU scores, and the x-axis represents steps ranging from 0 to 1000. The graph illustrates the performance improvement over steps for various configurations.]
Adaptation to New Language Pairs

- Distribute a new sub-network for new language pair and train the sub-network for fixed steps

LaSS reaches the bilingual model performance with fewer steps.
Adaptation to New Language Pairs

- Distribute a new sub-network for new language pair and train the sub-network for fixed steps

LaSS hardly drops on existing language pairs
Adaptation to New Language Pairs

- Distribute a new sub-network for new language pair and train the sub-network for fixed steps.

  easy adaptation is attributed to the language specific sub-network

  Only updates the corresponding parameters avoids catastrophic forgetting
Top/bottom layers prefer language specific capacity

The top deals with output projection layer and the bottom is related to embedding layer, which are both language-specific.
Mask similarity is positively correlated to language family

Similar languages tend to group together for both En→X and X→En

Learning Language Specific Sub-network for Multilingual Machine Translation [Lin et al., ACL 2021]
Summary for Multilingual Pre-training

• Multilingual fused pre-training
  – Training encoder on masked sequences composed of multiple language, concatenated or mixed words.

• Multilingual sequence-to-sequence pre-training
  – mBart: Recover original sentence from noised ones in multiple languages.
  – mRASP & mRASP2: augmenting data with randomly substitute of words from bilingual lexicon + monolingual reconstruction + contrastive learning
  – LaSS: use pre-training and fine-tuning to discover language-common sub-nets and language-specific sub-nets for MT
PART IV: Pre-training for Speech Translation
Speech-to-Text Translation (ST)

- source language *speech(audio)* → target lang *text*

**Application Type**
- (Non-streaming) ST e.g. video translation
- Streaming ST e.g. realtime conference translation

**System**
- Cascaded ST
- End-to-end ST
Cascaded ST System

- Challenges:
  1. Computationally inefficient
  2. Error propagation: Wrong transcription ➡ Wrong translation

ASR system

Speech ➡ Transcription

Good morning

MT system ➡ Translation

Bonjour

*do at this* and see if it works for you ➡ 这样做，看看它是否对你有用

*duet this* and see if it works for you ➡ 二重奏一下，看看它是否对你有用
• Single model to produce text translation from speech
• Basic model: Encoder-Decoder architecture (e.g. Transformer)
• Advantage:
  – Reduced latency, simpler deployment
  – Avoid error propagation

Basic Speech Translation Model (Same as MT)

Transformer-based: N-layer convolution + attention encoder, M-layer decoder

Training data: <audio seq., translation text>

Encoder

Decoder

Comment allez-vous ?

Beam Search

Add & Norm
Feed

Add & Norm
Multi-Head
Attention

CNN

fbank
log mel freq.

<BOSS>

Softmax

Decoder Layer

Decoder Layer

Decoder Layer

Softmax

Decoder Layer

Decoder Layer

Decoder Layer

Softmax

Decoder Layer

Decoder Layer

Decoder Layer

Add & Norm
Feed Forward

Add & Norm
Multi-Head
Attention

Add & Norm
Masked
Multi-Head
Attention

How are you ?
• Data scarcity - lack of large parallel audio-translation corpus
• Modality disparity between audio and text
• Performance gap of direct ST:
  – BLEU: ST 18.6 vs. MT 36.2 (on MuST-C En-De)
Pre-training for Speech Translation

- **MT Pre-training**
  - Decoder initialization from separately trained MT model
  - Single-modal(audio) Encoder-Decoder: COSTT [Dong et al, AAAI 2021b]

- **ASR Pre-training**
  - Curriculum Pre-training [Wang et al, ACL 2020]
  - LUT [Dong et al, AAAI 2021a]

- **Audio Pre-training**
  - Wav2vec & Wav2Vec2.0 [Schneider et al. Interspeech 2019, Baevski et al NeurIPS2020]

- **Raw Text Pre-training**
  - LUT [Dong et al, AAAI 2021a]

- **Bi-modal Pre-training**
  - TCEN-LSTM [Wang et al, AAAI 2020]
  - Chimera [Han et al, ACL 2021a]
  - XSTNet [Ye et al, Interspeech 2021]
  - Wav2vec2.0 + mBart + Self-training [Li et al, ACL 2021b]
  - FAT-ST [Zheng et al, ICML 2021]
How to use MT data with much larger scale to improve ST performance?
Separate Encoder-Decoder Pre-train

Speech Recognition
LibriSpeech corpus

Speech Translation
fine-tune on ST data

Machine Translation
WMT corpus

How are you?
Comment allez-vous?
Comment allez-vous?
Knowledge Distillation from MT model

MT pre-training $\rightarrow$ KL loss + ST Cross-entropy loss

End-to-End Speech Translation with Knowledge Distillation [Liu et al, Interspeech 2019]
How to make a single model’s decoder to perform text translation?

- Decoder $\rightarrow$ translation
- Encoder $\rightarrow$ Decoder $\rightarrow$ transcribe and translation

Pre-train ST’s decoder with full MT

Advantages of COSTT

- Unified training with both transcript and translation text
- Reduced encoder output size with CTC-guided shrinking
- Able to pre-train the decoder with external MT parallel data

Using external ASR data

How to use larger external ASR data to improve ST performance?

Dataset size
ST vs ASR

MuST-C ST data
ASR data

Dataset size (hours)

CommonVoice
LibriSpeech
Curriculum Pre-training with ASR data

1. ASR Cross entropy + ASR CTC loss
   I like to eat apple
   Transformer Decoder

2. Masked LM + Bilingual lexicon
   KL loss
   Ich esse gerne Apfel
   Transformer Encoder

3. Translation cross entropy
   Ich esse gerne Apfel
   Transformer Decoder

Curriculum Pre-training for End-to-end Speech Translation [Wang et al, ACL 2020]
ASR Pre-training helps ST

IWSLT & Librispeech

<table>
<thead>
<tr>
<th></th>
<th>En-De</th>
<th>En-Fr</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transformer ST</td>
<td>12.5</td>
<td>16.9</td>
</tr>
<tr>
<td>Transformer+ASR</td>
<td>13.1</td>
<td>18</td>
</tr>
<tr>
<td>Transformer+Curriculum</td>
<td>18.2</td>
<td>18.2</td>
</tr>
<tr>
<td>COSTT</td>
<td>18.6</td>
<td>18.2</td>
</tr>
</tbody>
</table>
Raw Text Pre-training

Using pre-trained LM in decoding weighting is easy!

But

🤔 How to use pre-trained BERT to improve ST performance?
Drawbacks of the Encoder-Decoder Structure

1. A single encoder is hard to capture the representation of audio for the translation.
2. Limited in utilizing the information of “transcription” in the training.
Motivation: Mimic human’s behavior

Question: How human translate?

“Listen-Understand-Translate” (LUT) model based motivated by human’s behavior
Motivation of Better Encoding

**Drawback 1:** A single encoder is not enough.

**Idea 1:** Introduce a semantic encoder

**Drawback 2:** Limit in using “transcript” info.

**Idea 2:** Utilizing the pre-trained representation (e.g. BERT) of the “transcript” to learn the semantic feature.
LUT: Utilizing Pre-trained Model on Raw Text

Training data: triples of

<speech, transcript_text, translate_text>

Transcript (z):
“Good morning”

Input (x):
Log-mel fbank feature

Acoustic Encoder (Listen)

Semantic Encoder (Understand)

Translation Decoder (Translate)

BERT representation

CTC loss

Distance loss

Translation(y):
“Bonjour”

CE loss

ST Benefits from BERT, with Raw Text Pre-training

IWSLT & Librispeech

Transformer ST
Transformer+ASR
Transformer+Curriculum
COSTT
LUT

Listen, Understand and Translate [Dong et al, AAAI 2021]
How to use larger raw audio data to improve ST performance?
Wav2Vec: Self-supervised Speech Representation Learning

Training data: LibriSpeech 960 hrs audio only

Minimize contrastive loss

\[ L = - \sum \left( \log \sigma(z_{t+1} \cdot h_t) + \sum \log \sigma(-z_{-} \cdot h_t) \right) \]

Bring closer context and acoustic state

Bring further context and negative sampled acoustic state

Low level acoustic state \( h \), each frame ~ 30ms, stride10ms

High-level context state \( c \), each frame ~ 210ms, stride10ms

wav2vec: Unsupervised Pre-training for Speech Recognition [Schneider et al, Interspeech 2010]
Wav2Vec2.0: Contrastive on quantized acoustic state

Quantized low-level acoustic state, each frame ~ 25ms, stride 20ms

Masked context during training

Training data: (audio only)
LibriSpeech 960 hrs
LibriVox 53k hrs

Minimize contrastive loss
\[ L = - \sum \log \frac{\exp \text{Sim}(c_t, q_t)}{\sum \exp \text{Sim}(c_t, q_{-})} + \text{penalty} \]

Bring closer masked context and quantized acoustic state

Speech Translation with Audio-Pretrain

Wav2vec Pretrain + Fine-tune on ST

Comment allez-vous ?

Decoder

Encoder

Wav2vec 2.0

How are you ?

MuST-C ST results

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>En-De</td>
<td>22.8</td>
<td>23.6</td>
<td>29.8</td>
<td>33.3</td>
</tr>
<tr>
<td>En-Fr</td>
<td>22.2</td>
<td>22.4</td>
<td>27.8</td>
<td>34.6</td>
</tr>
<tr>
<td>En-Ru</td>
<td>15.1</td>
<td>17</td>
<td>18.2</td>
<td>23.6</td>
</tr>
<tr>
<td>En-Ro</td>
<td>27.8</td>
<td>27.8</td>
<td>27.8</td>
<td>29.8</td>
</tr>
</tbody>
</table>

Self-training with Audio data

Step 0. Audio-only pre-training for Wav2vec2.0

Step 1. Freeze Wav2vec2.0, train on ST

Step 2. Self-train on generated pseudo-translation with LibriVox audio

CoVoST2 Results

<table>
<thead>
<tr>
<th>Language Pair</th>
<th>BLEU Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>En-De</td>
<td>20.2</td>
</tr>
<tr>
<td>En-Ca</td>
<td>21.8</td>
</tr>
<tr>
<td>En-Ar</td>
<td>23.8</td>
</tr>
<tr>
<td>En-Tr</td>
<td>26.5</td>
</tr>
</tbody>
</table>

[1] CoVoST 2 and Massively Multilingual Speech-to-Text Translation, [Wang et al InterSpeech 2021]
Bimodal Pre-training with Audio & MT data

• Chimera: Learning Fixed-size Shared Space for both audio and text, audio+MT pretraining [Han et al. 2021]
• XSTNet: Bring speech sequence to roughly similar length to text, then Pre-training & progressive multi-task fine-tuning [Ye et al. 2021]
• Wav2vec2.0-mTransformer LNA: Use both audio pertaining + multilingual pertained language model, and selective efficient fine-tuning [Li et al. ACL 2021]
• FAT-ST: Masked pre-training for fused audio and text [Zheng et al. ICML 2021]
Bi-modal Encoding Architecture for ST

Challenges: gap between text and audio
1. Length: ~20 (text) vs. ~ 1k-10k (audio)
2. Embedding space disparity
Insights from Cognitive Neuroscience

Speech and text interfere with each other in brain[1]

activation map
Convergence sites of speech (blue) and text (yellow)


Idea: Bridging the Speech-Text modality gap with Shared Semantic Representation

ST triple data:

<speech, transcript_text, translate_text>

Text Input
“everyone loves cats”

Audio Input

Translation
“tut le monde aime les chats”

Learning Shared Semantic Space for Speech-to-Text Translation [Chi Han, Mingxuan Wang, Heng Ji, Lei Li, Findings of ACL 2021]
Chimera Model for ST

Training with auxiliary objectives: ST + MT + Contrastive loss
Benefit: able to exploit large external MT data

Learning Shared Semantic Space for Speech-to-Text Translation Listen [Chi Han, Mingxuan Wang, Heng Ji, Lei Li, Findings of ACL 2021]
Chimera achieves the best (so far) BLEU on all languages in MuST-C

<table>
<thead>
<tr>
<th>Model</th>
<th>External Data</th>
<th></th>
<th>MuST-C</th>
<th>EN-X</th>
<th>EN-ES</th>
<th>EN-RO</th>
<th>EN-PT</th>
<th>EN-NL</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Speech</td>
<td>ASR</td>
<td>MT</td>
<td>EN-DE</td>
<td>EN-FR</td>
<td>EN-RU</td>
<td>EN-EN</td>
<td>EN-IT</td>
</tr>
<tr>
<td>FairSeq ST †</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>22.7</td>
<td>32.9</td>
<td>15.3</td>
<td>27.2</td>
<td>22.7</td>
</tr>
<tr>
<td>Espnet ST ‡</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>22.9</td>
<td>32.8</td>
<td>15.8</td>
<td>28.0</td>
<td>23.8</td>
</tr>
<tr>
<td>AFS *</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>22.4</td>
<td>31.6</td>
<td>14.7</td>
<td>26.9</td>
<td>23.0</td>
</tr>
<tr>
<td>Dual-Decoder ◊</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>23.6</td>
<td>33.5</td>
<td>15.2</td>
<td>28.1</td>
<td>24.2</td>
</tr>
<tr>
<td>STATST ‡</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>23.1</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>MAML †</td>
<td>×</td>
<td>×</td>
<td>✓</td>
<td>22.1</td>
<td>34.1</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Self-Training ◊</td>
<td>✓</td>
<td>✓</td>
<td>×</td>
<td>25.2</td>
<td>34.5</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>W2V2-Transformer *</td>
<td>✓</td>
<td>×</td>
<td>×</td>
<td>22.3</td>
<td>34.3</td>
<td>15.8</td>
<td>28.7</td>
<td>24.2</td>
</tr>
<tr>
<td>Chimera Mem-16</td>
<td>✓</td>
<td>×</td>
<td>✓</td>
<td>25.6</td>
<td>35.0</td>
<td>16.7</td>
<td>30.2</td>
<td>24.0</td>
</tr>
<tr>
<td>Chimera</td>
<td>✓</td>
<td>×</td>
<td>✓</td>
<td>27.1 *</td>
<td>35.6</td>
<td>17.4</td>
<td>30.6</td>
<td>25.0</td>
</tr>
</tbody>
</table>

Learning Shared Semantic Space for Speech-to-Text Translation [Chi Han, Mingxuan Wang, Heng Ji, Lei Li, Findings of ACL 2021]
End-to-end Speech Translation via Cross-modal Progressive Training [Rong Ye, Mingxuan Wang, Lei Li, Interspeech 2021]
Supports to train MT data

- Transformer MT model
- We can add more external MT data to train Transformer encoder & decoder
Supports inputs of two modalities

- Wav2vec2.0 [1] as the acoustic encoder
- We add two convolution layers with 2-stride to shrink the length.

---

Language indicator strategy

- We use language indicators to distinguish different tasks.

<table>
<thead>
<tr>
<th>Tasks</th>
<th>Source input</th>
<th>Target output</th>
</tr>
</thead>
<tbody>
<tr>
<td>MT</td>
<td>&lt;en&gt; This is a book.</td>
<td>&lt;fr&gt; c'est un livre.</td>
</tr>
<tr>
<td>ASR</td>
<td>&lt;audio&gt; <img src="AudioWaveform.png" alt="Audio Waveform" /></td>
<td>&lt;en&gt; This is a book.</td>
</tr>
<tr>
<td>ST</td>
<td>&lt;audio&gt; <img src="AudioWaveform.png" alt="Audio Waveform" /></td>
<td>&lt;fr&gt; c'est un livre.</td>
</tr>
</tbody>
</table>

End-to-end Speech Translation via Cross-modal Progressive Training [Rong Ye, Mingxuan Wang, Lei Li, Interspeech 2021]
Progressive Multi-task Training

Large-scale MT pre-training

Using external MT $D_{MT-ext}$

Multi-task Finetune

Using (1) external MT $D_{MT-ext}$
(2) $D_{ST}$ with <speech, translation>
(3) $D_{ASR}$ with <speech, transcript>

Progressive:
Don’t stop training $D_{MT-ext}$
XSTNet achieves State-of-the-art Performance

<table>
<thead>
<tr>
<th>Models</th>
<th>External Data</th>
<th>Pre-train Tasks</th>
<th>De</th>
<th>Es</th>
<th>Fr</th>
<th>It</th>
<th>Nl</th>
<th>Pt</th>
<th>Ro</th>
<th>Ru</th>
<th>Avg.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transformer ST [13]</td>
<td>×</td>
<td>ASR</td>
<td>22.8</td>
<td>27.4</td>
<td>33.3</td>
<td>22.9</td>
<td>27.2</td>
<td>28.7</td>
<td>22.2</td>
<td>15.1</td>
<td>24.9</td>
</tr>
<tr>
<td>AFS [31]</td>
<td>×</td>
<td>×</td>
<td>22.4</td>
<td>26.9</td>
<td>31.6</td>
<td>23.0</td>
<td>24.9</td>
<td>26.3</td>
<td>21.0</td>
<td>14.7</td>
<td>23.9</td>
</tr>
<tr>
<td>Dual-Decoder Transf. [15]</td>
<td>×</td>
<td>×</td>
<td>23.6</td>
<td>28.1</td>
<td>33.5</td>
<td>24.2</td>
<td>27.6</td>
<td>30.0</td>
<td>22.9</td>
<td>15.2</td>
<td>25.6</td>
</tr>
<tr>
<td>Tang et al. [2]</td>
<td>MT</td>
<td>ASR, MT</td>
<td>23.9</td>
<td>28.6</td>
<td>33.1</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>FAT-ST (Big) [6]</td>
<td>ASR, MT, mono-data†</td>
<td>FAT-MLM</td>
<td>25.5</td>
<td>30.8</td>
<td>-</td>
<td>-</td>
<td>30.1</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>W-Transf.</td>
<td>audio-only*</td>
<td>SSL*</td>
<td>23.6</td>
<td>28.4</td>
<td>34.6</td>
<td>24.0</td>
<td>29.0</td>
<td>29.6</td>
<td>22.4</td>
<td>14.4</td>
<td>25.7</td>
</tr>
<tr>
<td>XSTNet (Base)</td>
<td>audio-only*</td>
<td>SSL*</td>
<td>25.5</td>
<td>29.6</td>
<td>36.0</td>
<td>25.5</td>
<td>30.0</td>
<td>31.3</td>
<td>25.1</td>
<td>16.9</td>
<td>27.5</td>
</tr>
<tr>
<td>XSTNet (Expand)</td>
<td>MT, audio-only*</td>
<td>SSL*, MT</td>
<td>27.8</td>
<td>30.8</td>
<td>38.0</td>
<td>26.4</td>
<td>31.2</td>
<td>32.4</td>
<td>25.7</td>
<td>18.5</td>
<td>28.8</td>
</tr>
</tbody>
</table>

Table 1: Performance (case-sensitive detokenized BLEU) on MuST-C test sets. †: “Mono-data” means audio-only data from Librispeech, Libri-Light, and text-only data from Europarl/Wiki Text; *: “Audio-only” data from LibriSpeech is used in the pre-training of wav2vec2.0-base module, and “SSL” means the self-supervised learning from unlabeled audio data. ‡ uses OpenSubtitles as external MT data.

**XSTNet-Base**: Achieves the SOTA in the restricted setup

**XSTNet-Expand**: Goes better by using extra MT data
XSTNet better than cascaded ST! a gain of 2.6 BLEU

![Graph showing BLEU scores for different language pairs and models]

What is “Cascaded-Strong” system?

Cascaded-Strong is a system that combines a strong ASR model with large-scale MT data.

<table>
<thead>
<tr>
<th>Cascaded-Strong</th>
<th>Model</th>
<th>Training data</th>
<th>Performance (En-De)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASR</td>
<td>W2V2+ Transformer</td>
<td>MuST-C $D_{ASR}$</td>
<td>WER=13.0</td>
</tr>
<tr>
<td>MT</td>
<td>Transformer-base</td>
<td>WMT + MuST-C $D_{MT}$</td>
<td>BLEU=31.7</td>
</tr>
</tbody>
</table>
Audio and Multilingual Text Pretrain for Multilingual ST

- Encoder uses Wav2vec2.0 pre-trained on LibriVox-60k audio
- Decoder: mBart pre-trained on 50 monolingual text and 49 bitext
- ST finetune strategy (LNA):
  - Only fine-tune layer-norm and attention layers
- MT+ST multitask joint train with further parallel bitext data
Wav2vec2.0 retraining + Multilingual training effectively transfers to low resource source language

CoVoST2 Results

<table>
<thead>
<tr>
<th></th>
<th>BLEU</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fr-En</td>
<td>35.2</td>
</tr>
<tr>
<td>De-En</td>
<td>31.1</td>
</tr>
<tr>
<td>Es-En</td>
<td>27.6</td>
</tr>
<tr>
<td>Ca-En</td>
<td>24.1</td>
</tr>
</tbody>
</table>

Transformer
m-Transformer
Wav2vec2.0-mTransformer LNA

CoVoST2 Results

<table>
<thead>
<tr>
<th></th>
<th>BLEU</th>
</tr>
</thead>
<tbody>
<tr>
<td>En-De</td>
<td>30.9</td>
</tr>
<tr>
<td>En-Ca</td>
<td>22.3</td>
</tr>
<tr>
<td>En-Ar</td>
<td>18.0</td>
</tr>
<tr>
<td>En-Tr</td>
<td>17.0</td>
</tr>
<tr>
<td>En-Zh</td>
<td>28.2</td>
</tr>
</tbody>
</table>

Transformer
m-Transformer
Wav2vec2.0-mTransformer joint train

Multilingual Speech Translation with Efficient Finetuning of Pretrained Models [Li et al, ACL 2021]
Fused Acoustic and Text Masked Language Model (FAT-MLM)

Pre-training data:
1. Librispeech ASR 960h
2. Libri-light audio 3,748h
3. Europarl/wiki text 2.3M
4. MuST-C 408h
5. Europarl MT 1.9M

Transformer Encoder

L2 loss
Cross-entropy
Transformer Encoder
Cross-entropy

Acoustic embedding
2D Deconvolution

Transformer Encoder

Mask

2D Convolution

<s>[Mask]Morning</s>
<s>Guten</s>
FAT-ST

Training:
• Pre-train FAT-MLM with all data
• Init FAT-ST with FAT-MLM, decoder copy encoder
• Further fine-tune on MuST-C ST data.

\[
l_{ST}(s, y) \quad \text{and} \quad l_{MT}(x, y)
\]
Joint audio&text Pre-training task helps ST

<table>
<thead>
<tr>
<th>Pretrain Method</th>
<th>Models</th>
<th>En→De</th>
<th>En→Es</th>
<th>En→Nl</th>
<th>Avg.</th>
<th>Model Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>No Pretraining</td>
<td>ST</td>
<td>19.64</td>
<td>23.68</td>
<td>23.01</td>
<td>22.11</td>
<td>31.25M</td>
</tr>
<tr>
<td></td>
<td>ST + ASR</td>
<td>21.70</td>
<td>26.83</td>
<td>25.44</td>
<td>24.66 (+2.55)</td>
<td>44.82M</td>
</tr>
<tr>
<td></td>
<td>ST + ASR &amp; MT</td>
<td>21.58</td>
<td>26.37</td>
<td>26.17</td>
<td>24.71 (+2.60)</td>
<td>56.81M</td>
</tr>
<tr>
<td></td>
<td>ST + MAM</td>
<td>20.78</td>
<td>25.34</td>
<td>24.46</td>
<td>23.53 (+1.42)</td>
<td>33.15M</td>
</tr>
<tr>
<td></td>
<td>ST + MAM + ASR</td>
<td>22.41</td>
<td>26.89</td>
<td>26.49</td>
<td>25.26 (+3.15)</td>
<td>46.72M</td>
</tr>
<tr>
<td></td>
<td>Liu et al. (2020b)</td>
<td>22.55</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>Le et al. (2020)</td>
<td>23.63</td>
<td>28.12</td>
<td>27.55</td>
<td>26.43 (+4.32)</td>
<td>51.20M</td>
</tr>
<tr>
<td></td>
<td>Cascade§</td>
<td>23.65</td>
<td>28.68</td>
<td>27.91</td>
<td>26.75 (+4.64)</td>
<td>83.79M</td>
</tr>
<tr>
<td></td>
<td>FAT-ST (base)</td>
<td>22.70</td>
<td>27.86</td>
<td>27.03</td>
<td>25.86 (+3.75)</td>
<td>39.34M</td>
</tr>
<tr>
<td>ASR &amp; MT</td>
<td>ST</td>
<td>21.95</td>
<td>26.83</td>
<td>26.03</td>
<td>24.94 (+2.83)</td>
<td>31.25M</td>
</tr>
<tr>
<td></td>
<td>ST + ASR &amp; MT</td>
<td>22.05</td>
<td>26.95</td>
<td>26.15</td>
<td>25.05 (+2.94)</td>
<td>56.81M</td>
</tr>
<tr>
<td>MAM</td>
<td>FAT-ST (base)</td>
<td>22.29</td>
<td>27.21</td>
<td>26.26</td>
<td>25.25 (+3.14)</td>
<td>39.34M</td>
</tr>
<tr>
<td>FAT-MLM</td>
<td>FAT-ST (base)</td>
<td>23.68</td>
<td>28.61</td>
<td>27.84</td>
<td>26.71 (+4.60)</td>
<td>39.34M</td>
</tr>
<tr>
<td></td>
<td>FAT-ST (big)</td>
<td>23.64</td>
<td>29.00</td>
<td>27.64</td>
<td>26.76 (+4.65)</td>
<td>58.25M</td>
</tr>
</tbody>
</table>

Fused Acoustic and Text Encoding for Multimodal Bilingual Pretraining and Speech Translation, [Zheng et al ICML 2021]
Pre-training Improves ST Performance

• MuST-C Results

<table>
<thead>
<tr>
<th></th>
<th>Transformer-ST</th>
<th>FAT-ST</th>
<th>Chimera</th>
<th>XSTNet</th>
</tr>
</thead>
<tbody>
<tr>
<td>En-De</td>
<td>27.8</td>
<td>27.4</td>
<td>30.8</td>
<td>38</td>
</tr>
<tr>
<td>En-Es</td>
<td>27.4</td>
<td>30.8</td>
<td>33.3</td>
<td>31.2</td>
</tr>
<tr>
<td>En-Fr</td>
<td>38</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>En-Nl</td>
<td></td>
<td>27.2</td>
<td></td>
<td>31.2</td>
</tr>
<tr>
<td>En-Ru</td>
<td></td>
<td></td>
<td></td>
<td>18.4</td>
</tr>
<tr>
<td></td>
<td>Direct Supervision</td>
<td>Contrastive</td>
<td>Masked LM</td>
<td>Knowledge distillation</td>
</tr>
<tr>
<td>---------------------</td>
<td>--------------------</td>
<td>-------------</td>
<td>-----------</td>
<td>------------------------</td>
</tr>
<tr>
<td>MT Parallel Text</td>
<td>COSTT</td>
<td></td>
<td></td>
<td>[Liu et al. 2019]</td>
</tr>
<tr>
<td>ASR Speech-Transcript</td>
<td>LUT</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Audio-only</td>
<td>Wav2vec</td>
<td>Wav2vec 2.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Raw text</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Speech+Text</td>
<td>Chimera</td>
<td>FAT-ST</td>
<td></td>
<td>XSTNet</td>
</tr>
</tbody>
</table>
Summary for Speech Translation Pre-training

- Parallel speech translation data is scarce
- Pre-training to utilize external large data
  - MT data (Parallel text)
  - ASR data (Speech-transcript)
  - Raw text (Monolingual and Multilingual)
  - Audio-only
- Network architecture to solve modality disparity
  - CNN-Transformer
  - Fixed-size shared memory module
  - Bimodal input with length shrinking for audio
- Techniques to better pre-train and better fine-tune
  - Contrastive prediction
  - Masked LM
  - Quantization of audio representation
  - Knowledge distillation
  - Progressive pre-training
Summary

• Basics
  – NMT, Transformer encoder decoder.
  – Pre-training paradigm for NLP
• Monolingual Pre-training for NMT
  – Encoder pre-training
  – Seq-to-seq pre-training
• Multilingual Pre-training for NMT
• Pre-training for Speech Translation
Thanks

• Rong Ye, Chi Han, Qianqian Dong for help on beautification of the slides.
Reference

- Monolingual Pre-training
  - When and Why are Pre-trained Word Embeddings Useful for Neural Machine Translation [Qi et al., NAACL 2018]
  - Improve Neural Machine Translation by Building Word Vector [Zhang et al., AI 2020]
  - A bag of useful tricks for practical neural machine translation: Embedding layer initialization and large batch size [Neishi et al, ACL 2017]
  - Unsupervised pretraining for sequence to sequence learning, [Ramachandran et al., EMNLP 2017]
  - Incorporate BERT into Neural Machine Translation, [Zhu et al ICLR 2020]
  - Acquiring Knowledge from Pre-trained Model to Neural Machine Translation, [Weng et al AAAI 2020]
  - Towards Making Most of BERT for NMT, [Yang et al AAAI 2020]
  - Comparison between Pre-training and Large-scale Back-translation, [Huang et al., ACL 2021]
  - MASS: Pre-train for Sequence to Sequence Generation, [Song et al ICML 2019]
  - BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension, [Lewis et al ACL 2020]
Multilingual Pre-training

- Cross-lingual Language Model Pre-training [Conneau et al NeurIPS 2019]
- Alternating Language Modeling for Cross-Lingual Pre-Training [Yang et al AAAI 2020]
- mBART: Multilingual Denoising Pre-training for Neural Machine Translation [Liu et al., TACL 2020]
- Pre-training Multilingual Neural Machine Translation by Leveraging Alignment Information [Lin et al., EMNLP 2020]
- CSP: Code-Switching Pre-training for Neural Machine Translation [Yang et al., EMNLP 2020]
- Contrastive Learning for Many-to-many Multilingual Neural Machine Translation [Pan et al., ACL 2021]
- Learning Language Specific Sub-network for Multilingual Machine Translation [Lin et al., ACL 2021]
Reference

• Speech Translation
  – wav2vec: Unsupervised Pre-training for Speech Recognition
  – wav2vec 2.0: A framework for self-supervised learning of speech representations
  – Investigating self-supervised pre-training for end-to-end speech translation
  – Self-supervised representations improve end-to-end speech translation (wav2vec + LSTM seq2seq)
  – Large-Scale Self-and Semi-Supervised Learning for Speech Translation
  – Consecutive Decoding for Speech-to-text Translation
  – “Listen, Understand and Translate”: Triple Supervision Decouples End-to-end Speech-to-text Translation
  – Learning Shared Semantic Space for Speech-to-Text Translation [ACL 21]
  – Multilingual Speech Translation with Efficient Finetuning of Pretrained Models [ACL 21]
  – Fused Acoustic and Text Encoding for Multimodal Bilingual Pretraining and Speech Translation [ICML 21]
  – End-to-end Speech Translation via Cross-modal Progressive Training [Interspeech 21]
  – Curriculum Pre-training for End-to-end Speech Translation [ACL 20]
  – End-to-End Speech Translation with Knowledge Distillation [Interspeech 19]