Exploring Massively Multilingual Neural Machine Translation

White paper video : Orhan Firat

Notes

Goal of this paper : Universal translation model.

* Building a single translation model for whole language pair translation and increasing the translation quality across the board.
* This paper covers 103 languages.(en->any and any->en).
* Total amount of data used : 25 billion (from 103 languages)
* Multilingual NMT is basically a transfer mechanism .it transfers knowledge learnt from high resource language to low resource language.(positive transfer)
* Amount of data plays a major role in neural machine translation.Low amount of data leads to memorization.Language pair containing mid amount of data tries to generalize well.High resouce language pairs usually generalize well and quality is near human-level.

Trainability and optimization

***** Training a larger model is not a big task.how to deploy this model during production is a big question.Techniques such as network pruning,vocabulary reduction etc.. helps.

Additional papers which includes more details about trainability and optimization:

* Training Deeper NMT models with transparent attention.
* Massively multilingual neural machine translation.
* Adaptive scheduling for multitask learning.

Training:

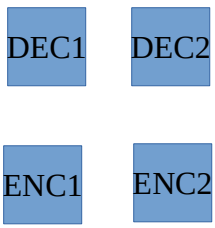
Data:

* If the data is imbalanced,after a certain training,low reource language starts to overfit and high resource language error is high.
*The common strategies used for balancing the data is upsampling/oversampling , temperature tuning.

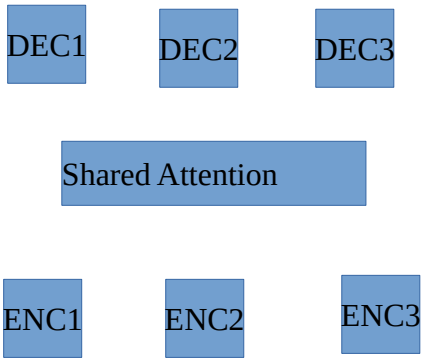
Model:

* For training multilingual models,first we need to scale up the size of neural network.
* Different model types for MNMT.

1) Shared embeddings but different encoders ,decoders and attentions



2) Shared attention but different encoders and decoders.



3) sharing everything .

In this paper,they followed third architecture ,sharing everything.There are 103 languages and scripts are different .Commonlity between some languages are low.sharing everything leads to interference.

Model related research question :

* Do deep or wide networks drive greater quality gains?
* How deep or wide should we go ?
* what quality boost do we obtain by sacrificing half of the tasks (for eg : en->any)

Architectures used for training in this paper listed below :

- Transformer bilingual baseline (400 M)
-Transformer (12 layers , 8k wide ,10 heads)
-Transformer (48 layers ,8k wide,16 heads) -1.3Billion parameters
-Transformer(24 layers,16k wide,32 heads) – 1.3 Billion
-Transformer(128 layers,16k wide,32 heads) – 6 billion

According to their experimental results,deeper models transfer and generalize better compared to wider models.
Wider models memorize a lot due to high embedding dimension.
Its better to go deeper than wider.

Execution:
Training such large architecture is not a trivial task.Training pipeline parallelism framework details are explained in “GPIPE – pipeline parallelism framework paper”

As we decided ,going deeper generalize well.how deeper one can go ?.There are so many things to play with model architecture.

So we have Mixture of expert model.

There are multiple experts within the model.

Transformer architecture consisters of 6 encoer stack.each encoder stack contains Multihead attention layer followed by Feed forward layer.Replaced feedforward layer with dispatch.Dispatch is connected to mixture of experts via sparse gate connection.Dispatch receives the tokens and decides which MOE to receive .

Dispatch main task is to route or sub-route tokens to specific experts .Token level optimization leads to high throughput.
Special case of MOE - it conditions on language tags(language IDS).

Adding MOE increases model parameters sublinearly .MOE increase model capacity.

TransformerMOE – (12 layers,128 experts) – 10 billion parameters
TransformerMOE-(12 layers,512 experts) – 50 billion parameters.

Additional cost item is introduced which balances across each experts to utilize all the experts.

Main takeaways from this paper:

* Deeper models generalize better.
* Interference is a bigger problem.scaling doesnt help to solve interference problem.
* Multilingual transformer model which shares everything amplifying the problem of interference.

Compute and Machine learning systems:

* Trained model on 1024 TPU v3 chips
* GPIPE
*Bloat16 (Brain floating point)
*Rematerialization (gradient checkpointing)
*Large batches (4M)
* sub linear scaling

Challenges and open problems :

* Better learning objectives.
* how to add new tasks and languages?
* Unsupervised MT
- M4 as a knowledge base for unsupervised MT
- Adapting to unseen languages(modalities)
* Parallel transfer – smarter scheduler,mitigating interference

* serial transfer – maximize transfer without forgetting .

Additional papers:

* Massively multilingual neural machine translation in the wild – findings and challenges.
* Evaluating the cross-lingual effectiveness of massively MNMT.
* Investigating MNMT representation at scale.
* Simple scalable adaptation for NMT.
* Outrageously large neural networks
* Drawing the map of languages:
  Investigating Multilingual NMT representations at scale.
* Missing ingredients in zero shot neural machine translation .