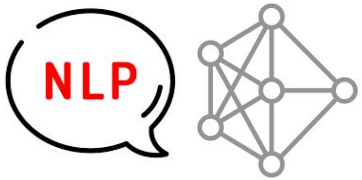
CHULA ENGINEERING
Foundation toward Innovation
COMPUTER



Neural Machine Translation

2110572: Natural Language Processing Systems

Peerapon Vateekul & Ekapol Chuangsuwanich
Department of Computer Engineering,
Faculty of Engineering, Chulalongkorn University

Credit:

- Kasidis Kanwatchara
- Can Udomcharoenchaikit & Nattachai Tretasayuth

Outline

- Part1) MT models
  - mBART (2020)
  - NLLB (started 2018)
    - m2m-100 (2020)
    - NLLB-200 (2022)
- Part2) MT data sets
  - WMT
  - OPUS
  - FLORES-200
- Part3) Evaluation
  - Accuracy-based score (BLEU, charF++)
  - Model-based score
    - COMET
    - Quality Estimation

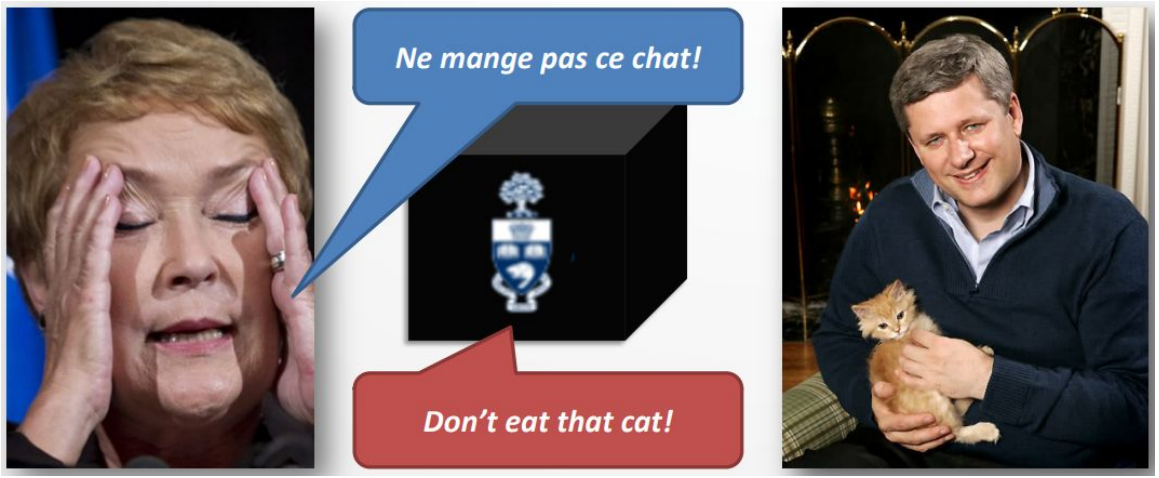
+

Part1) MT models

- mBART (2020)
- NLLB (started 2018)
  - m2m-100 (2020)
  - NLLB-200 (2022)

Introduction

Machine Translation (MT) is a research field in NLP that aims to create a system that can translate text from one language to another.

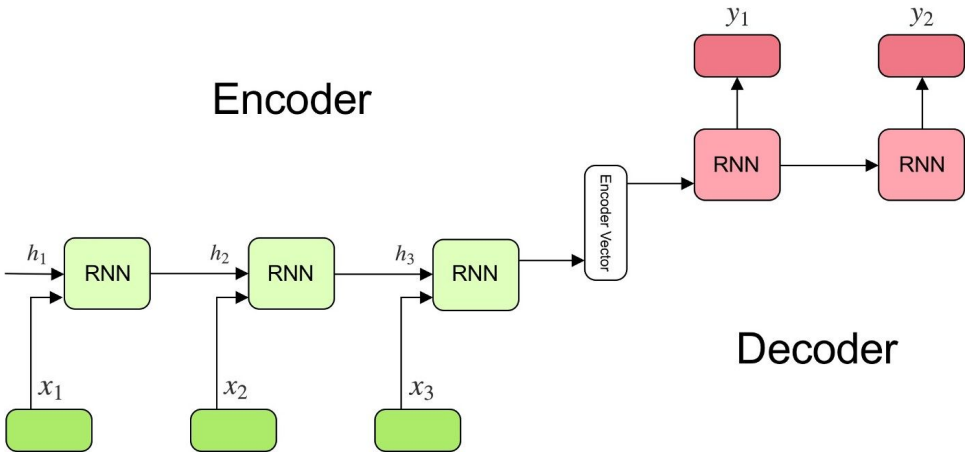


Development of Machine Translation

1. Rule-based Machine Translation
   a. Manual, hand-crafted rules for vocabulary, grammar, etc.
   b. Low quality translation and time consuming.
   c. Cannot utilize context information!
2. Statistical Machine Translation
   a. Use statistics from a parallel corpus
   b. Google translate (from 2006-2016)
3. Neural Machine Translation
   a. Competitive performance but hard to debug
   b. Google translate (now)

Neural Machine Translation (NMT)

Prior to the Transformer, the dominant model in NMT was RNN-based encoder-decoder.



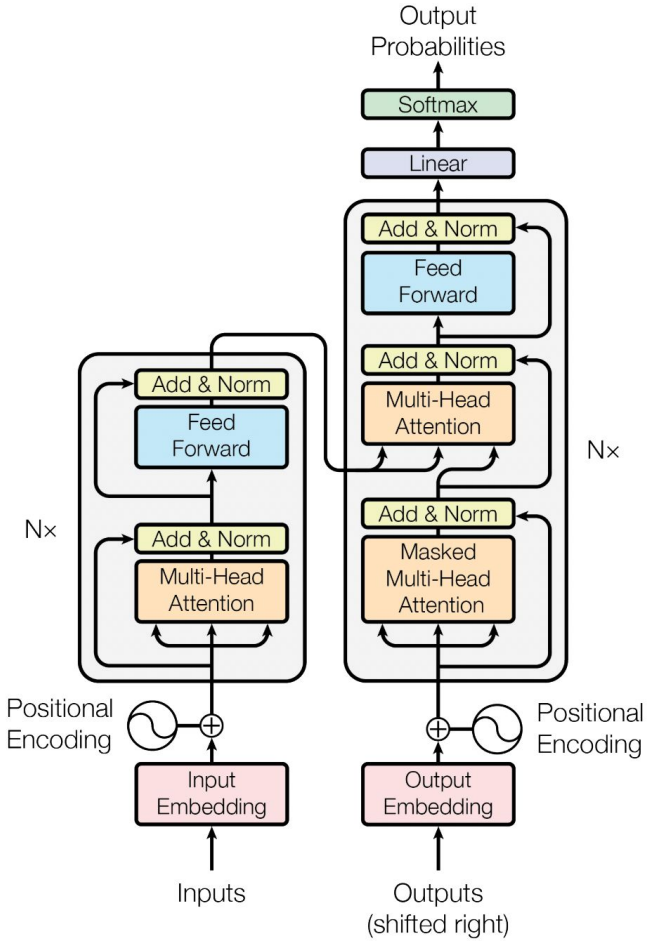
Transformer

The self-attention mechanism in the Transformer allows for better input representations and faster computation due to parallelization.

This brings big performance improvements and thus allows researchers to scale the model more efficiently.

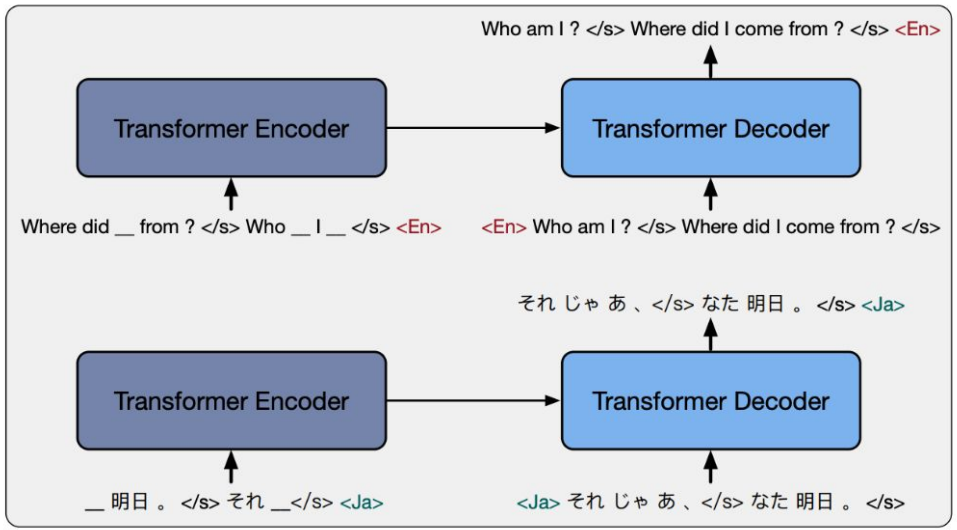
Table 2: The Transformer achieves better BLEU scores than previous state-of-the-art models on the English-to-German and English-to-French newstest2014 tests at a fraction of the training cost.

Model	BLEU		Training Cost (FLOPs)	
	EN-DE	EN-FR	EN-DE	EN-FR
ByteNet [18]	23.75			
Deep-Att + PosUnk [39]		39.2		$1.0 \cdot 10^{20}$
GNMT + RL [38]	24.6	39.92	$2.3 \cdot 10^{19}$	$1.4 \cdot 10^{20}$
ConvS2S [9]	25.16	40.46	$9.6 \cdot 10^{18}$	$1.5 \cdot 10^{20}$
MoE [32]	26.03	40.56	$2.0 \cdot 10^{19}$	$1.2 \cdot 10^{20}$
Deep-Att + PosUnk Ensemble [39]		40.4		$8.0 \cdot 10^{20}$
GNMT + RL Ensemble [38]	26.30	41.16	$1.8 \cdot 10^{20}$	$1.1 \cdot 10^{21}$
ConvS2S Ensemble [9]	26.36	**41.29**	$7.7 \cdot 10^{19}$	$1.2 \cdot 10^{21}$
Transformer (base model)	27.3	38.1	**$3.3 \cdot 10^{18}$**	
Transformer (big)	**28.4**	**41.8**	$2.3 \cdot 10^{19}$	



1) mBART

A standard encoder-decoder transformer model that pretrains on input denoising objective. There are two steps: (1) pretrain on the denoising task and (2) finetune on the MT task.

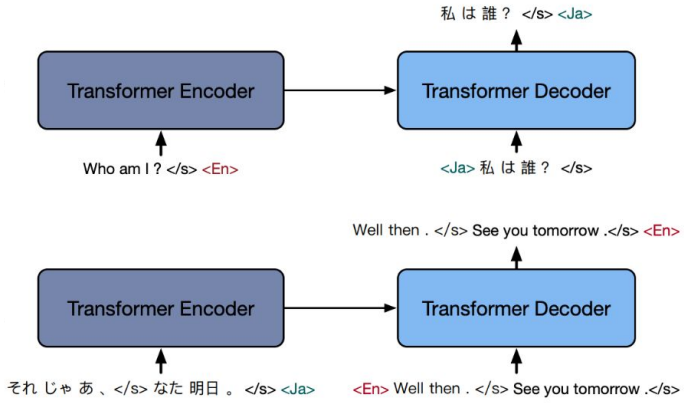


Multilingual Denoising Pre-Training (mBART)

Multilingual Denoising Pre-training for Neural Machine Translation. Liu et al 2020

1) mBART (cont.)

Pretraining on multilingual data (~1TB) improve translation quality especially low-resource language pairs.



Fine-tuning on Machine Translation

Languages	En-Gu		En-Kk		En-Vi		En-Tr		En-Ja		En-Ko	
Data Source	WMT19		WMT19		IWSLT15		WMT17		IWSLT17		IWSLT17	
Size	10K		91K		133K		207K		223K		230K	
Direction	←	→	←	→	←	→	←	→	←	→	←	→
Random	0.0	0.0	0.8	0.2	23.6	24.8	12.2	9.5	10.4	12.3	15.3	16.3
mBART25	**0.3**	**0.1**	**7.4**	**2.5**	**36.1**	**35.4**	**22.5**	**17.8**	**19.1**	**19.4**	**24.6**	**22.6**
Languages	En-Nl		En-Ar		En-It		En-My		En-Ne		En-Ro	
Data Source	IWSLT17		IWSLT17		IWSLT17		WAT19		FLoRes		WMT16	
Size	237K		250K		250K		259K		564K		608K	
Direction	←	→	←	→	←	→	←	→	←	→	←	→
Random	34.6	29.3	27.5	16.9	31.7	28.0	23.3	34.9	7.6	4.3	34.0	34.3
mBART25	**43.3**	**34.8**	**37.6**	**21.6**	**39.8**	**34.0**	**28.3**	**36.9**	**14.5**	**7.4**	**37.8**	**37.7**
Languages	En-Si		En-Hi		En-Et		En-Lt		En-Fi		En-Lv	
Data Source	FLoRes		ITTB		WMT18		WMT19		WMT17		WMT17	
Size	647K		1.56M		1.94M		2.11M		2.66M		4.50M	
Direction	←	→	←	→	←	→	←	→	←	→	←	→
Random	7.2	1.2	10.9	14.2	22.6	17.9	18.1	12.1	21.8	20.2	15.6	12.9
mBART25	**13.7**	**3.3**	**23.5**	**20.8**	**27.8**	**21.4**	**22.4**	**15.3**	**28.5**	**22.4**	**19.3**	**15.9**

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).

Multilingual Denoising Pre-training for Neural Machine Translation. Liu et al 2020

1) mBART (cont.)

It can also do zero-shot MT (with some tricks) that yields decent results.

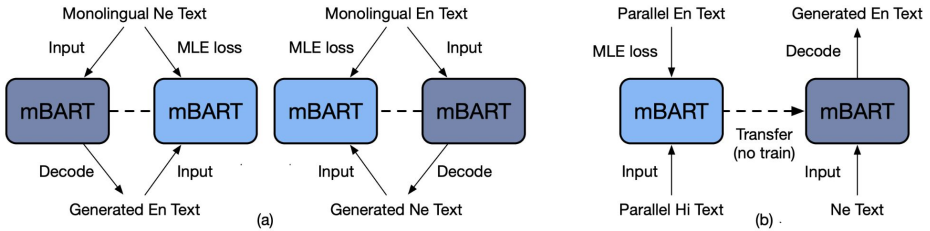


Figure 5: Illustrated frameworks for unsupervised machine translation via (a) back-translation (b) language transfer where Ne-En is used as an example. For both cases, we initialize from multilingual pre-training (e.g. mBART25).

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**

Table 10: **Unsupervised MT via Back-Translation**. En-De, En-Ro are initialized by mBART02, while En-Ne, En-Si are initialized by mBART25. Our models are trained on monolingual data used in pre-training.

Multilingual Denoising Pre-training for Neural Machine Translation. Liu et al 2020

1) mBART (cont.)

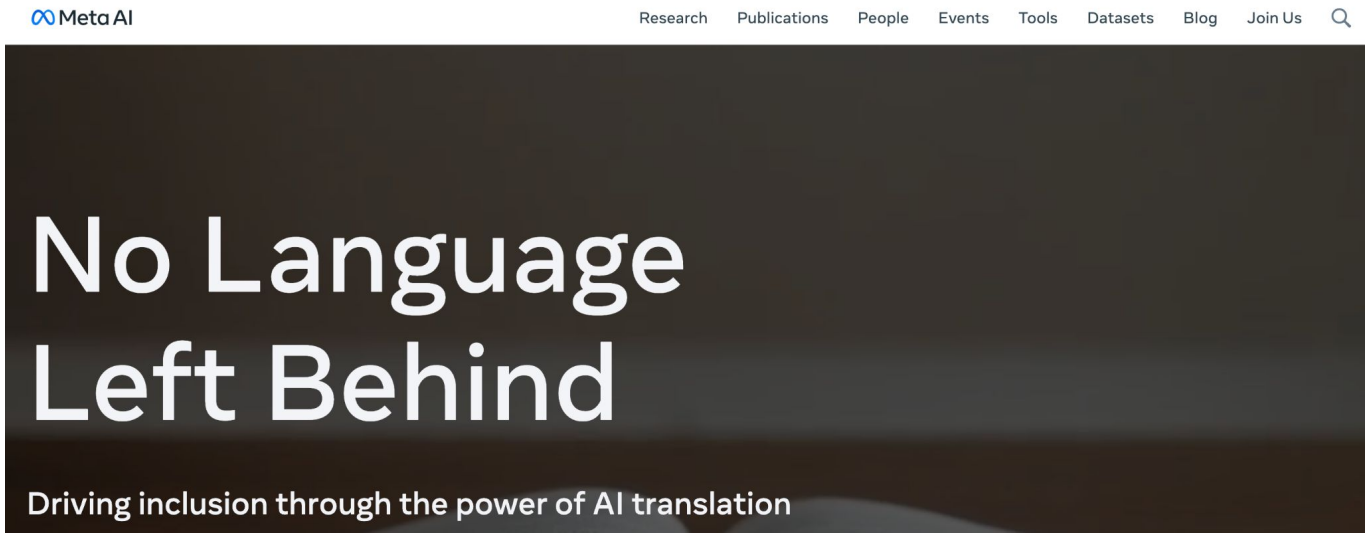
- Originally trained on 25 languages, 25 more languages (50 total languages) were added via continual pretraining.
- Thai language included! The model is very large though.

Data size	Languages
10M+	German, Czech, French, Japanese, Spanish, Russian, Polish, Chinese
1M - 10M	Finnish, Latvian, Lithuanian, Hindi, Estonian
100k to 1M	Tamil, Romanian, Pashto, Sinhala, Malayalam, Dutch, Nepali, Italian, Arabic, Korean, Hebrew, Turkish, Khmer, Farsi, Vietnamese, Croatian, Ukrainian
10K to 100K	Thai, Indonesian, Swedish, Portuguese, Xhosa, Afrikaans, Kazakh, Urdu, Macedonian, Telugu, Slovenian, Burmese, Georgia
10K-	Marathi, Gujarati, Mongolian, Azerbaijani, Bengali

Multilingual Translation with Extensible Multilingual Pretraining and Finetuning. Tang et al 2020

2) No Language Left Behind (NLLB) [FB started in 2018]

From the webpage - “No Language Left Behind (NLLB) is a first-of-its-kind, AI breakthrough project that open-sources models capable of delivering evaluated, high-quality translations **directly between 200 languages**”...



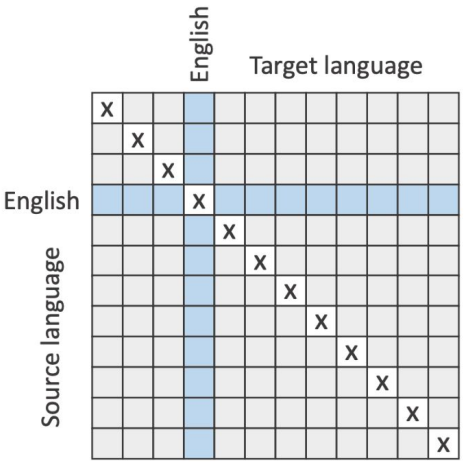
https://ai.facebook.com/research/no-language-left-behind

No Language Left Behind

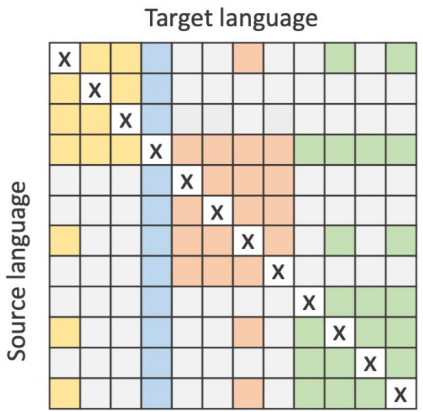
2.1) m2m-100

Previous works mostly focus on translation from/to English (English-centric).

This paper introduces a many-to-many translation model and dataset of 100 languages (that's 9900 directions!)



(a) English-Centric Multilingual



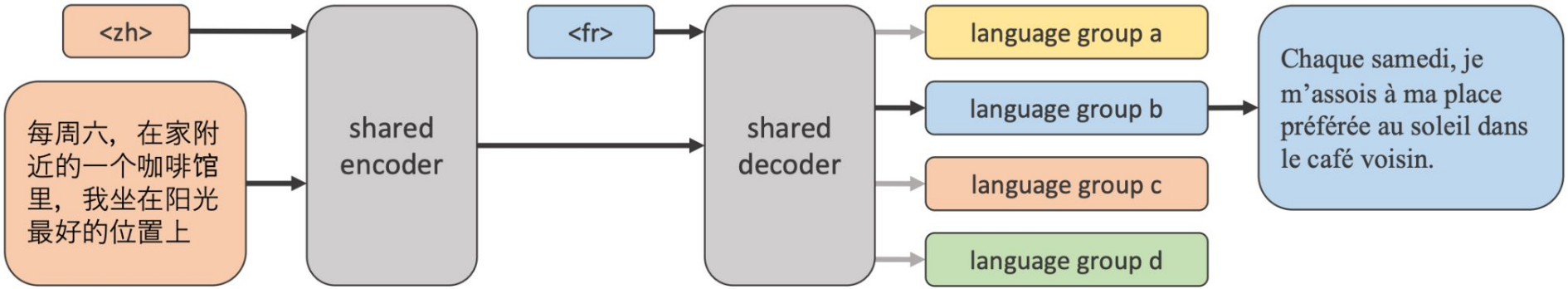
(b) M2M-100: Many-to-Many Multilingual Model

Beyond English-Centric Multilingual Machine Translation. Fan et al 2020

No Language Left Behind

2.1) m2m-100 (cont.)

The model is also an encoder-decoder model with additional language-specific sparse models.



(c) Translating from Chinese to French with Dense + Language-Specific Sparse Model

Beyond English-Centric Multilingual Machine Translation. Fan et al 2020

No Language Left Behind

2.1) m2m-100 (cont.)

The model outperforms mBART even though it trains on 100 languages. (Thai language is also covered by this model)

Benchmark	Model	BLEU
mBART	Previous Work (Liu et al., 2020)	23.9
	M2M-100	**24.6**
CCMatrix	Previous Work (Schwenk et al., 2019)	16.3
	M2M-100	**18.7**
OPUS100	Previous Work (Zhang et al., 2020)	14.1
	M2M-100	**18.4**

Table 11: **Comparison on various evaluation settings from previous work**. We display the best performing model from the published work and report average BLEU on the test set. For these comparisons, we use the tokenization and BLEU evaluation script used by each work for comparability. Liu et al. (2020) report Low/Mid resource directions into and out of English and High resource directions into English, we average across all. Schwenk et al. (2019) report the full matrix on 28 languages, we average across all. Zhang et al. (2020) report results on non-English directions, we average across all.

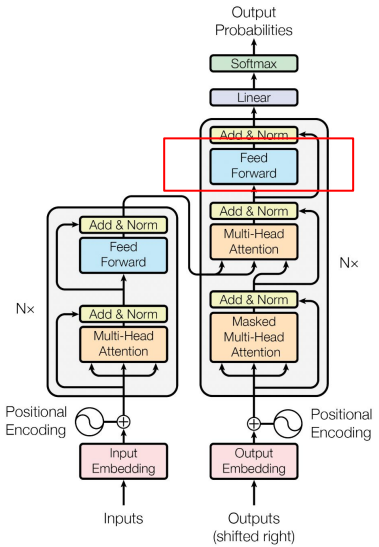
Beyond English-Centric Multilingual Machine Translation. Fan et al 2020

2.2) NLLB-200

No Language Left Behind

This model is capable of a total of 40,602 translation directions!

It is also an encoder-decoder transformer model. However, it is a sparse model (Mixture of Experts transformer).



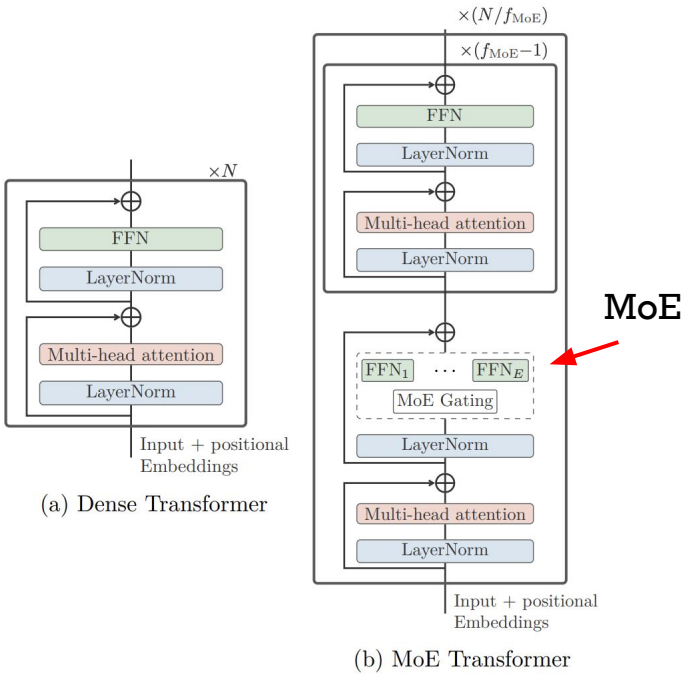


Figure 16: **Illustration of a Transformer encoder with MoE layers** inserted at a $1{:}f_{\text{MoE}}$ frequency. Each MoE layer has E experts and a gating network responsible for dispatching tokens.

No Language Left Behind: Scaling Human-Centered Machine Translation. NLLB Team 2022

No Language Left Behind

2.2) NLLB-200 (cont.)

With just 1.3B parameters, it outperforms Google Translate on average.

(Distilled) weight available!
https://huggingface.co/facebook/nllb-200-distilled-600M

	eng_Latn-xx	xx-eng_Latn	xx-yy	Avg.
87 languages				
M2M-100	-/-	-/-	-/-	13.6/-
Deepnet	-/-	-/-	-/-	18.6/-
NLLB-200	**35.4**/52.1	**42.4**/62.1	**25.2**/43.2	**25.5**/43.5
101 languages				
DeltaLM	26.6/-	33.2/-	16.4/-	16.7/-
NLLB-200	**34.0**/50.6	**41.2**/60.9	**23.7**/41.4	**24.0**/41.7

Table 30: **Comparison on FLORES-101 devtest**. We evaluate over full FLORES-101 10k directions. We report both spBLEU/chrF++ where available. All spBLEU numbers are computed with FLORES-101 SPM tokenizer. Scores for DeltaLM are taken from FLORES-101 leaderboard. M2M-100 and Deepnet average is only over 87 languages that overlap with FLORES-101, we also show NLLB-200 performance on that subset of languages. NLLB-200 outperforms previous state of the art models by a significant margin, even after supporting twice as many languages.

	eng_Latn-xx		xx-eng_Latn		Average	
	low	v.low	low	v.low	low	v.low
Google Translate	**32.3/50.3**	**27.0/46.5**	35.9/57.1	35.8/57.0	34.1/53.7	31.3/51.7
NLLB-200	30.3/48.2	25.7/45.0	**41.3/60.4**	**41.1/60.3**	**35.8/54.3**	**33.4/52.6**

Table 34: **Comparison on 102 Low-Resource Directions on FLORES-200 devtest against commercial translation systems.** We evaluate on all English-centric low-resource directions that overlap between FLORES-200 and Google's Translation API as of this writing. We report both spBLEU/chrF++ and bold the best score. We observe that NLLB-200 outperforms significantly on xx-eng_Latn and overall average.

No Language Left Behind

2.2) NLLB-200 (cont.)

This paper also explain how they created a parallel corpus, including a multilingual sentence encoder and a language identification model. (The whole paper is 192 pages)

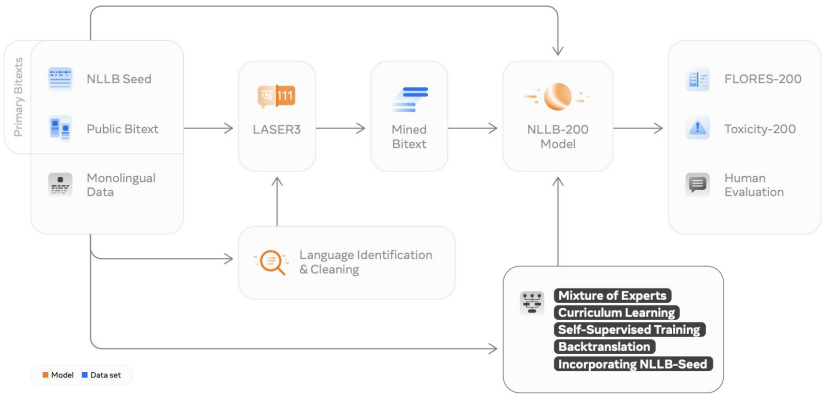


Figure 14: **Modeling Contributions of No Language Left Behind**: As highlighted, we describe several modeling techniques to enable coverage of hundreds of languages in one model. We focus on effectively scaling model capacity while mitigating overfitting, as well as how to improve backtranslation for low-resource languages and incorporate NLLB-SEED.

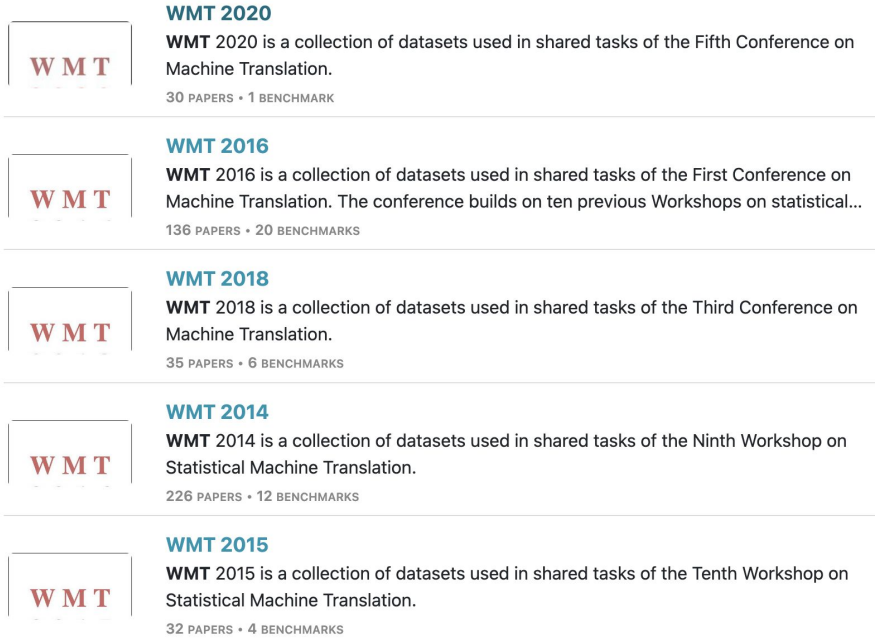
No Language Left Behind: Scaling Human-Centered Machine Translation. NLLB Team 2022

Part2) Notable datasets

WMT
OPUS
FLORES-200

1) WMT

WMT (Workshop on Statistical Machine Translation) - This is a machine translation dataset composed from a collection of various sources, including (1) news commentaries and (2) parliament proceedings.



https://www.statmt.org/wmt14/translation-task.html

2) OPUS

Opus - https://opus.nlpl.eu/

... the open parallel corpus

OPUS is a growing collection of translated texts from the web. In the OPUS project we try to convert and align free online data, to add linguistic annotation, and to provide the community with a publicly available parallel corpus. OPUS is based on open source products and the corpus is also delivered as an open content package. We used several tools to compile the current collection. All pre-processing is done automatically. No manual corrections have been carried out.

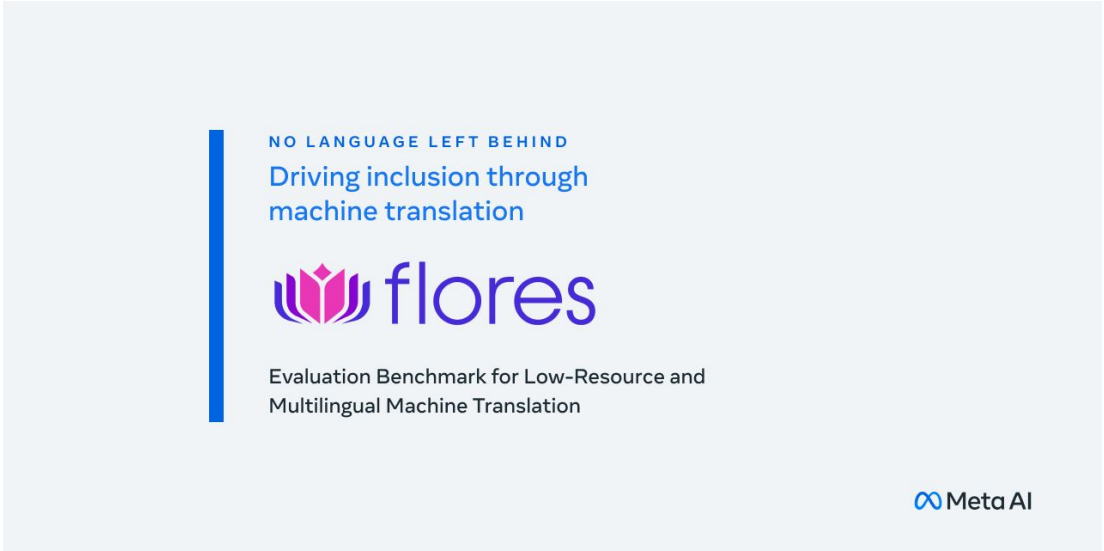
The OPUS collection is growing! Check this page from time to time to see new data arriving ...
Contributions are very welcome! Please contact <jorg.tiedemann@helsinki.fi >

Search & download resources: -- select -- -- select -- all show all versions

No Language Left Behind

3) FLORES-200

FLORES-200 - consists of translations from 842 distinct web articles, totaling 3001 sentences. These sentences are divided into three splits: dev, devtest, and test (hidden). On average, sentences are approximately 21 words long.



https://github.com/facebookresearch/flores/blob/main/flores200/README.md

Part3) Evaluating a Translation

Accuracy-based score (BLEU, charF++)

Model-based score

- COMET
- Quality Estimation

Evaluation

1. Human Judgement
   a. expensive, slow, non-reproducible (different judges – different biases).

2. Automatic Evaluation

   2.1) Accuracy-based score

   a. **BLEU** (BiLingual Evaluation Understudy) - Most popular
   b. METEOR
   c. **chrF** - May be more suitable for Thai language

   2.2) Model-based score - better correlation with human judgement but requires training.

   d. **With reference: COMET**
   e. **Without reference: Quality Estimation**

Human Evaluation

Human judgements of MT quality usually come in the form of segment-level scores, such as:

1. Human-mediated Translation Edit Rate (HTER) [1]
   a. the MT outputs are manually corrected (#edit)
   b. then the original outputs are compared to the edited ones by computing TER.

$$\text{TER} = \frac{\text{\# of edits}}{\text{average \# of reference words}}$$

2. Direct Assessment (DA)
   a. A quality score (satisfaction score) of 0 to 100 is given for a translation by human

3. Multidimensional Quality Metrics (MQM)

	A	B	C	D	E	F	G	H
1	MQM Scorecard: Top-Level Error Typology with 4 Severity Levels							
2								
3		*Error Severity Levels:*	Neutral	Minor	Major	Critical	Error Type Penalty Total	
4		*Severity Penalty Multipliers:*	0	1	5	25		
5	ET Nos	Error Types	Error Counts				ET Weights	ETPTs
6	1	Terminology	2	7	7	0	1.0	42.0
7	2	Accuracy	4	14	7	1	1.0	74.0
8	3	Linguistic conventions	1	23	9	0	1.0	68.0

[1] https://www.cs.umd.ed

https://themqm.org/error-types-2/1_scorecards/values-and-scores/

	A	B	C	D	E	F	G	H
1	MQM Scorecard: Top-Level Error Typology with 4 Severity Levels							
2								
3		*Error Severity Levels:*	Neutral	Minor	Major	Critical	Error Type Penalty Total	
4		*Severity Penalty Multipliers:*	0	1	5	25		
5	ET Nos	Error Types	Error Counts				ET Weights	ETPTs
6	1	Terminology	2	7	7	0	1.0	42.0
7	2	Accuracy	4	14	7	1	1.0	74.0
8	3	Linguistic conventions	1	23	9	0	1.0	68.0
9	4	Style	5	7	3	0	1.0	22.0
10	5	Locale convention	1	12	5	0	1.0	37.0
11	6	Audience appropriateness	0	2	1	0	1.0	7.0
12	7	Design and markup	0	6	1	0	1.0	11.0
13	8	Custom						
14							Absolute Penalty Total:	261.00
15								
16		Evaluation Word Count:	10184				Per-Word Penalty Total:	0.0256
17		Reference Word Count:	1000				Overall Normed Penalty Total:	25.63
18		Scaling Parameter (SP):	1.00				Overall Quality Score:	97.44
19		Max. Score Value:	100.00					
20		Threshold Value:	85.00				Pass/Fail Rating:	Pass

Figure 1: Scorecard with Top-Level MQM Error Types

	A	B	C	D	E	F	G	H
1	MQM Scorecard: Top-Level Error Typology with 4 Severity Levels							
2								
3		Error Severity Levels:	Neutral	Minor	Major	Critical	Error Type Penalty Total	
4		Severity Penalty Multipliers:	0	1	5	25		
5	ET Nos	Error Types	Error Counts				ET Weights	ETPTs
6	1	Terminology	2	7	7	0	1.0	42.0
7	2	Accuracy	4	14	7	1	1.0	74.0
8	3	Linguistic conventions	1	23	9	0	1.0	68.0
9	4	Style	5	7	3	0	1.0	22.0
10	5	Locale convention	1	12	5	0	1.0	37.0
11	6	Audience appropriateness	0	2	1	0	1.0	7.0
12	7	Design and markup	0	6	1	0	1.0	11.0
13	8	Custom						
14							Absolute Penalty Total:	261.00
15								
16		Evaluation Word Count:	10184				Per-Word Penalty Total:	0.0256
17		Reference Word Count:	1000				Overall Normed Penalty Total:	25.63
18		Scaling Parameter (SP):	1.00				Overall Quality Score:	97.44
19		Max. Score Value:	100.00					
20		Threshold Value:	85.00				Pass/Fail Rating:	Pass

Figure 1: Scorecard with Top-Level MQM Error Types

The scorecard can be confusing on first glance. It is important to understand the different values entered on the scorecard, how an evaluator uses the card, and how the math functions

Accuracy-Based Evaluation

1) Evaluation - BLEU (BiLingual Evaluation Understudy)

The most popular metric to (try to) measure the quality of predicted translations.

The idea is to measure the similarity of the predictions with human references.

> *The BLEU metric ranges from 0 to 1. Few translations will attain a score of 1 unless they are identical to a reference translation. For this reason, even a human translator will not necessarily score 1. […] on a test corpus of about 500 sentences (40 general news stories), a human translator scored 0.3468 against four references and scored 0.2571 against two references.*

— BLEU: a Method for Automatic Evaluation of Machine Translation, 2002.

Automatic Evaluation - BLEU

The score is calculated based on **clipped n-gram precision.**

Clipping prevents repetitive predicted sentence to get a good score

For example, we could have

- **Target Sentence:** He eats an apple
- **Predicted Sentence:** He He He

This means that the 1-gram precision is 3/3 or 100%.

https://towardsdatascience.com/foundations-of-nlp-explained-bleu-score-and-wer-metrics-1a5ba06d812b

Automatic Evaluation - BLEU

The score is calculated based on **clipped n-gram precision.**

So we limit the count of each word to the maximum of times that the word occurs in the Target Sentence

"He" occurs max. one time in 2 reference sentences. So we clip it to 1.

- **Target Sentence 1:** He eats a sweet apple
- **Target Sentence 2:** He is eating a tasty apple
- **Predicted Sentence:** He He He eats tasty fruit

Word	*Matching Sentence*	*Matched Predicted Count*	*Clipped Count*
He	*Both*	*3*	*1*
eats	*Target 1*	*1*	*1*
tasty	*Target 2*	*1*	*1*
fruit	*None*	*0*	*0*
Total		*5*	*3*

Now the 1-gram precision becomes 3/6
There are 6 words in the predicted sentence

Note that precision now refers to the clipped precision

https://towardsdatascience.com/foundations-of-nlp-explained-bleu-score-and-wer-metrics-1a5ba06d812b

Automatic Evaluation - BLEU

The score is calculated based on **clipped n-gram precision.**

Now we calculate the n-gram (clipped) precision for all N. The widely used number for N is 4 and a uniform weight w_n of N/4.

Let's look at all the 2-grams in our predicted sentence:

Target Sentence: The guard arrived late because it was raining

Predicted Sentence: The guard arrived late because of the rain

$$\begin{aligned} Geometric\ Average\ Precision\ (N) &= exp(\sum_{n=1}^{N} w_n\ log\ p_n) \\ &= \prod_{n=1}^{N} p_n^{w_n} \\ &= (p_1)^{\frac{1}{4}} \cdot (p_2)^{\frac{1}{4}} \cdot (p_3)^{\frac{1}{4}} \cdot (p_4)^{\frac{1}{4}} \end{aligned}$$

The precision of 2-grams is 4/7. (#correct/#total)

[pre(1-gram)*pre(2-gram)*pre(3-gram)*pre(4-gram)]$^{1/4}$

https://towardsdatascience.com/foundations-of-nlp-explained-bleu-score-and-wer-metrics-1a5ba06d812b

Automatic Evaluation - BLEU

The score is calculated based on **clipped n-gram precision.**

Next, we compute the brevity penalty. This **penalizes predicted sentences that are too short.**

For example, a predicted sentence can contain only one word and get a perfect n-gram precision score

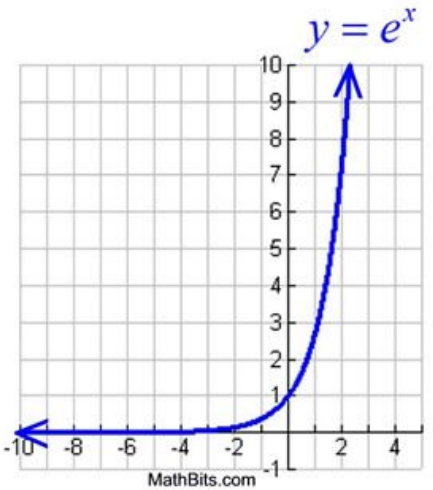
$$Brevity\ Penalty = \begin{cases} 1, & \text{if } c > r \\ e^{(1-r/c)}, & \text{if } c <= r \end{cases}$$

[0,1]

1

[0,1]

- c is *predicted length = number of words in the predicted sentence* and
- r is *target length = number of words in the target sentence*



https://towardsdatascience.com/foundations-of-nlp-explained-bleu-score-and-wer-metrics-1a5ba06d812b

Automatic Evaluation - BLEU

The score is calculated based on **clipped n-gram precision.**

Finally, we can compute the BLEU score by

$$Bleu\ (N) = Brevity\ Penalty \cdot Geometric\ Average\ Precision\ Scores\ (N)$$

2
(larger is better)

1
(larger is better)

https://towardsdatascience.com/foundations-of-nlp-explained-bleu-score-and-wer-metrics-1a5ba06d812b

Automatic Evaluation - BLEU

The score is calculated based on **clipped n-gram precision.**

Finally, we can compute the BLEU score by

$$Bleu\ (N) = Brevity\ Penalty \cdot Geometric\ Average\ Precision\ Scores\ (N)$$

Even though BLEU score is widely used, it has some important weaknesses:

- It does not consider words that have the **same meaning** to be correct. For example, for the word "dog", we can either use "หมา" or "สุนัข".
- It ignores the **importance of words**. With Bleu Score an incorrect word like "to" or "an" that is less relevant to the sentence is penalized just as heavily as a word that contributes significantly to the meaning of the sentence.
- Most importantly, **higher BLEU does not always mean a good score based on human judgement [1].**

https://towardsdatascience.com/foundations-of-nlp-explained-bleu-score-and-wer-metrics-1a5ba06d812b

[1] https://aclanthology.org/J18-3002.pdf

2) Automatic Evaluation - chrF++

- A score based on **character n-gram precision and recall.**
  - No tokenization required!
- The score averages over all n-grams.
  - The widely used number is 6 characters
- Later, *word* n-gram (2-gram) is added to the metric since it correlates more strongly with human judgement.
  - Now tokenization is required..

$$\text{CHRF}\beta = (1+\beta^2)\frac{\text{CHRP}\cdot\text{CHRR}}{\beta^2\cdot\text{CHRP}+\text{CHRR}}$$

where CHRP and CHRR stand for character n-gram precision and recall arithmetically averaged over all n-grams:

- CHRP
  percentage of n-grams in the hypothesis which have a counterpart in the reference;
- CHRR
  percentage of character n-grams in the reference which are also present in the hypothesis.

and β is a parameter which assigns β times more importance to recall than to precision – if $\beta = 1$, they have the same importance.

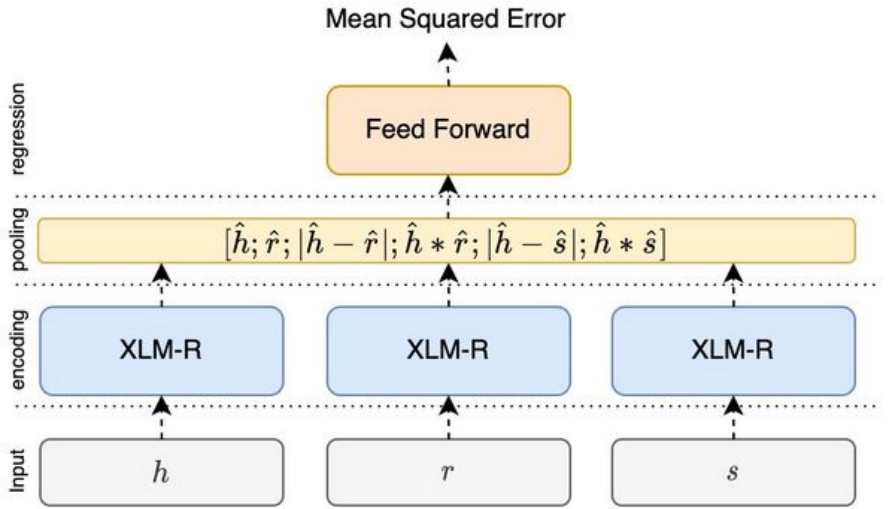
Recent experiments have shown that adding word 1-grams and 2-grams to the standard character 6-grams improves the Pearson correlation with direct human assessments. If you want to use only character n-grams, just set the word n-gram order to 0.

https://github.com/m-popovic/chrF
https://aclanthology.org/W15-3049.pdf

Model-based evaluation (COMET with reference)

COMET (with reference): variation 1

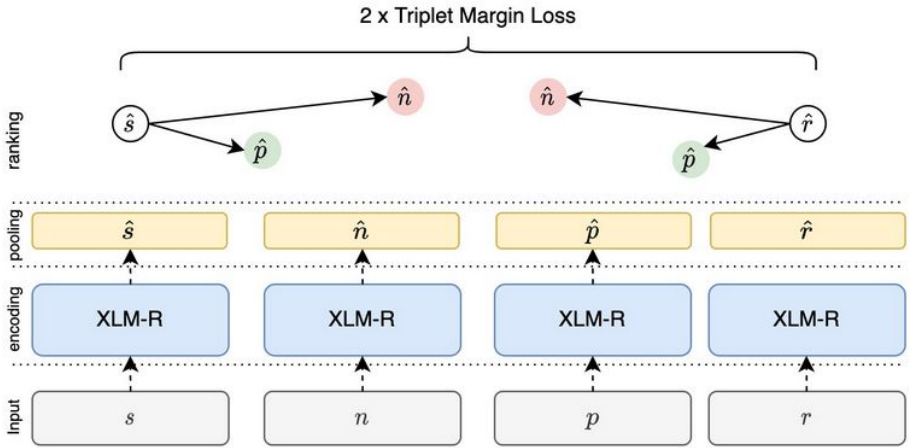
- Given a hypothesis (prediction) *h*, a reference (answer) *r*, and a source *s* as inputs
- COMET uses a multilingual encoder (XLM-R) to extract the features from the inputs.
- Concatenate them and feed it to a feed-forward regressor.
- The target score can be anything from human such as HTER or DA.



https://github.com/Unbabel/COMET

COMET (with reference): variation 2

- With a different architecture, it can also rank two different translations
- Given a worse translation *n*, a better translation *p*, a reference (answer) *r*, and a source *s* as inputs
- COMET uses a multilingual encoder (XLM-R) to extract the features from the inputs.
- Optimizes on the triplet loss.



https://github.com/Unbabel/COMET

+ Quality Estimation
What if we don't have reference (answer)?

Quality Estimation

An actively researched field where one attempt to **estimate the quality of a translation** *without* access to a reference translation.

This is a particularly hard task since neural networks are known to often be confident while giving wrong answers.

	Direct Assessment												
Encoder	**km-en**	**ps-en**	**en-ja**	**en-cs**	**en-mr**	**ru-en**	**ro-en**	**en-zh**	**en-de**	**et-en**	**si-en**	**ne-en**	**avg.**
Baseline (Zerva et al., 2021)													
XLM-R	0.615	0.601	0.295	0.535	0.419	0.703	0.828	0.513	0.500	0.806	0.565	0.793	0.598
Pretrained models													
InfoXLM	0.619	0.603	0.328	0.510	0.462	0.731	0.829	0.554	0.516	0.803	0.561	0.777	0.608
RemBERT	0.600	0.621	0.338	0.525	0.447	0.680	0.818	0.487	0.491	0.810	0.525	0.747	0.591
XLM-R	0.610	0.579	0.325	0.503	0.405	0.715	0.832	0.541	0.514	0.782	0.540	0.740	0.591
Sentence-level only													
XLM-R	0.628	0.591	0.350	0.531	0.551	0.761	0.859	0.577	0.568	0.800	0.565	0.796	0.631
InfoXLM	0.629	0.623	0.348	0.515	0.574	0.747	0.858	0.586	0.551	0.828	0.568	0.790	0.635
RemBERT	0.634	0.631	0.346	0.570	0.564	0.754	0.862	0.534	0.531	0.822	0.550	0.782	0.632
Few-shot Language Adaptation													
XLM-R	0.650	0.619	0.352	0.551	0.546	0.753	0.852	0.571	0.554	0.813	0.562	0.798	0.635
InfoXLM	0.641	0.650	0.367	0.549	0.549	0.751	0.855	0.591	0.565	0.824	0.563	0.803	0.642
RemBERT	0.625	0.641	0.367	0.568	0.563	0.756	0.857	0.540	0.527	0.824	0.568	0.796	0.636
Sentence + word-level training													
InfoXLM	0.617	0.586	0.344	0.532	0.572	0.761	0.865	0.586	0.579	0.829	0.576	0.804	0.637
RemBERT	0.634	0.628	0.356	0.564	0.571	0.762	0.860	0.541	0.553	0.826	0.564	0.799	0.638
Few-shot Language Adaptation													
InfoXLM	0.643	0.632	0.335	0.557	0.560	0.766	0.860	0.575	0.582	0.833	0.578	0.809	0.644
RemBERT	0.644	0.645	0.356	0.567	0.568	0.759	0.856	0.545	0.552	0.835	0.561	0.804	0.641
Final Ensemble													
Ensemble 6x	**0.664**	**0.669**	**0.380**	**0.591**	**0.593**	**0.782**	**0.871**	**0.597**	**0.593**	**0.845**	**0.588**	**0.820**	**0.666**

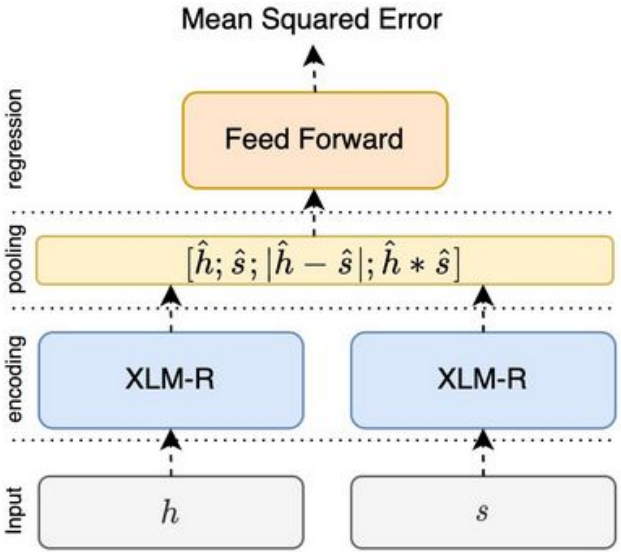
Even the best system does not exactly correlate with human judgement

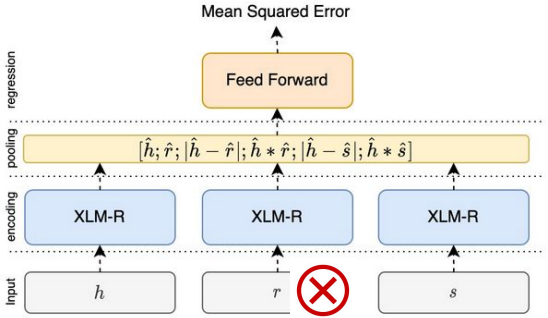
Table 1: Results for sentence-level QE in terms of Spearman correlation for DA.

https://github.com/Unbabel/COMET

COMET (without reference)

- **Everything is the same with the first variation of COMET with reference except you don't give it a reference text.**
- Given a hypothesis *h* and a source *s* as inputs
- COMET uses a multilingual encoder (XLM-R) to extract the features from the inputs.
- Concatenate them and feed it to a feed-forward regressor.
- The target score can be anything from human such as HTER or DA.





https://github.com/Unbabel/COMET

Evaluation - Conclusion

- A paper from Microsoft [1] advocates for COMET and ChrF.

Based on our findings, we suggest the following best practices for the use of automatic metrics:

1. Use a pretrained metric as the main automatic metric; we recommend COMET. Use a string-based metric for unsupported languages and as a secondary metric, for instance ChrF. Do not use BLEU, it is inferior to other metrics, and it has been overused.

[1] https://arxiv.org/pdf/2107.10821.pdf