# Machine Translation

## Introduction, Statistical NMT, Seq2Seq+Attention, Transformers, Multilingual, Evaluation and Non-Autoregressive Translation

Some slides borrowed from Anoop Kunchukuttan,
Images taken from the respective research papers

# Automatic conversion of text/speech

*Be the change you want to see in the world*

*वह परिवर्तन बनो जो संसार में देखना चाहते हो*



**Government:** administrative requirements, education, security.

**Enterprise:** product manuals, customer support

**Social:** travel (signboards, food), entertainment (books, movies, videos)

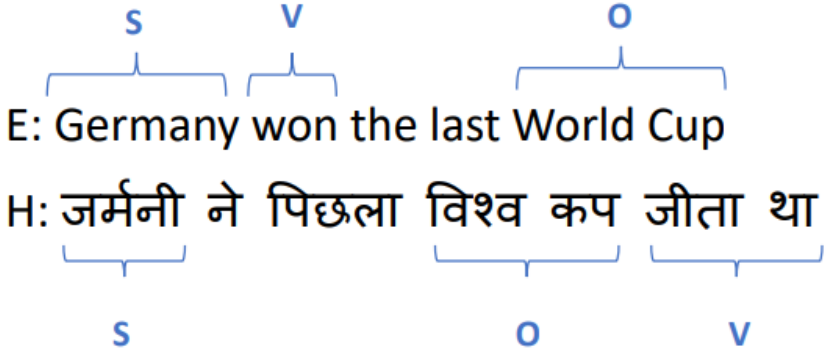
**Translation under the hood**

- Cross-lingual Search
- Cross-lingual Summarization
- Building multilingual dictionaries

*Any multilingual NLP system will involve some kind of machine translation at some level*

# What is Machine Translation?

**Word order: SOV (Hindi), SVO (English)**




**Free (Hindi) vs rigid (English) word order**

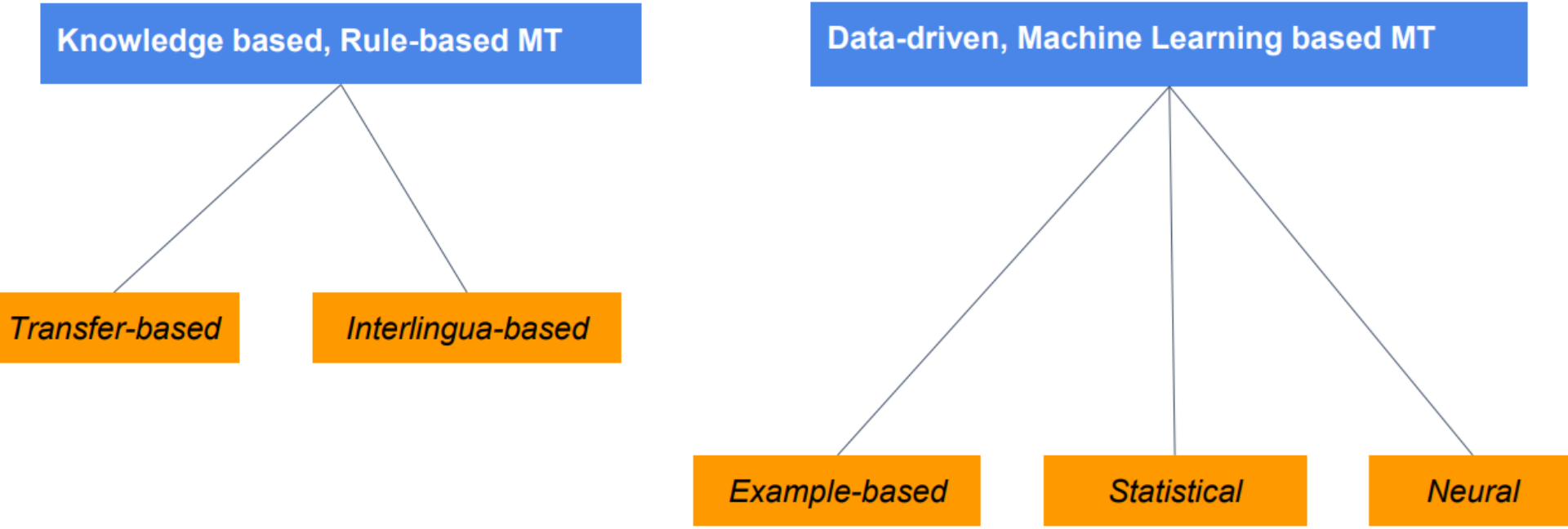
पिछला विश्व कप जर्मनी ने जीता था *(correct)*

The last World Cup Germany won *(grammatically incorrect)*
The last World Cup won Germany *(meaning changes)*

*Language Divergence ➔ the great diversity among languages of the world*

*The central problem of MT is to bridge this language divergence*

# Approaches to build MT systems



# Statistical MT

| Parallel Corpus | |
|---|---|
| A boy is sitting in the kitchen | एक लडका रसोई मे बैठा है |
| A boy is playing tennis | एक लडका टेनिस खेल रहा है |
| A boy is sitting on a round table | एक लडका एक गोल मेज पर बैठा है |
| Some men are watching tennis | कुछ आदमी टेनिस देख रहे है |
| A girl is holding a black book | एक लडकी ने एक काली किताब पकडी है |
| Two men are watching a movie | दो आदमी चलचित्र देख रहे है |
| A woman is reading a book | एक औरत एक किताब पढ रही है |
| A woman is sitting in a red car | एक औरत एक काले कार मे बैठा है |

***Key Idea***

*Co-occurrence of translated words*

*Words which occur together in the parallel sentence are likely to be translations (higher P(f|e))*

# IBM Models

- IBM came up with a series of increasingly complex models
- Called Models 1 to 5
- Simper models used to initialize the complex models
- This pipelined training helped ensure better solutions

# Neural Machine Translation

# Encode-Attend-Decode models for NMT

- Neural Machine Translation By Jointly Learning to Align and Translate, Bahdanu et. al., 2015

- Augmented Encoder-Decoder with attention mechanism
- Perform Beam Search on Decoder to generate most plausible translation
- Evaluate using BLEU metric

- GNMT: Google Neural Machine Translation System, 2016
- 8-layer LSTM Encoder-Decoder
- 10 days training with 96 GPUs - achieves new SOTA

# Transformers for NMT

- Attention is all you need, Vaswani et. al., 2017
- Proposed as a much faster alternative to LSTM-based systems
- Reduces training time to 3.5 days with 8 GPUs!
- Took 1 full year to successfully use Transformers in other tasks
- BERT truly demonstrated the power of Transformers in NLP and beyond.

# How to effectively use monolingual data?

# Back-Translation

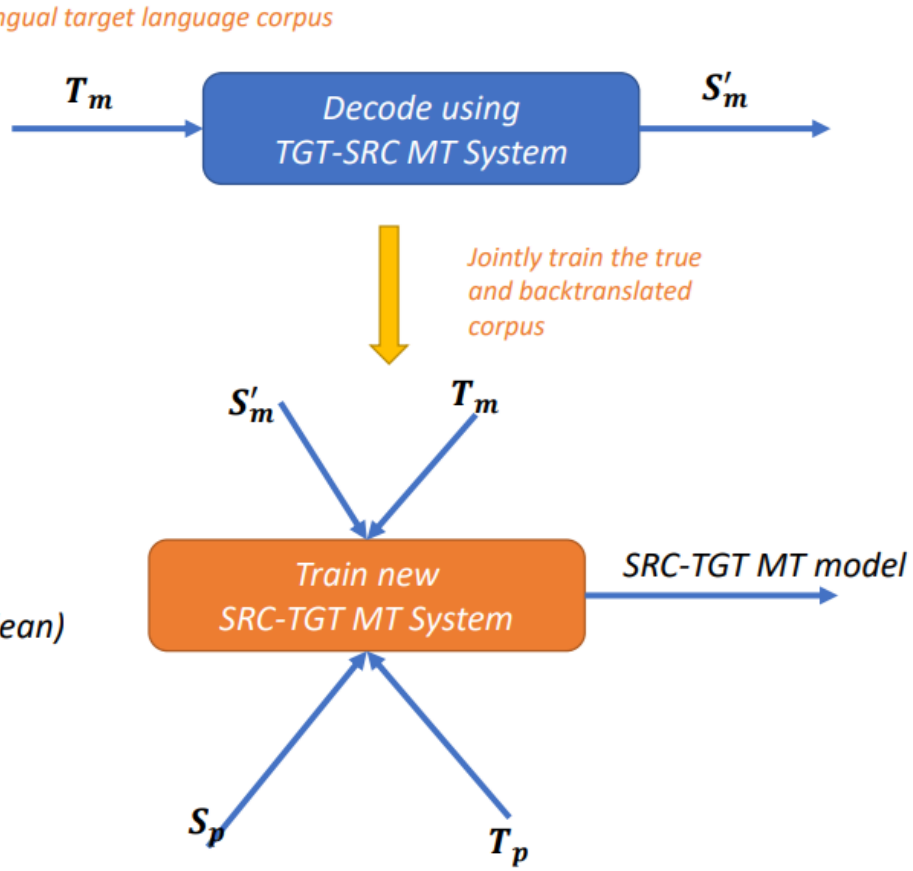
*Create pseudo-parallel corpus using Target to source model (Backtranslated corpus)*

*Need to find the right balance between true and backtranslated corpus*

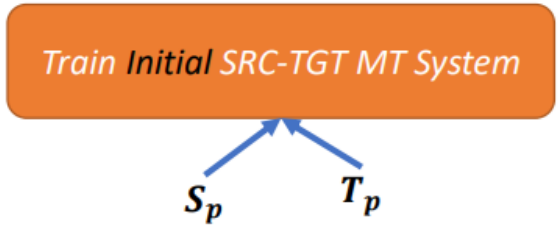
***Why is backtranslation useful?***

- *Target side language model improves (target side is clean)*
- *Adaptation to target language domain*
- *Prevent overfitting by exposure to diverse corpora*

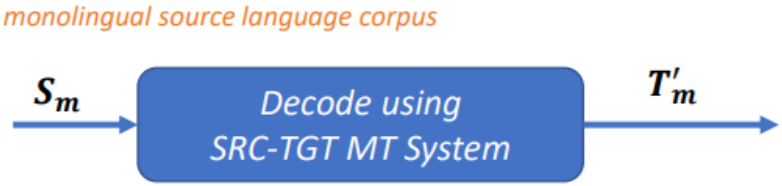
*Particularly useful for low-resource languages*



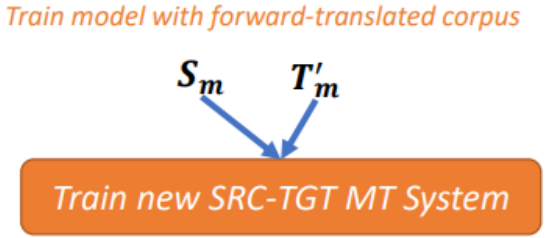
# Self Training




*Create pseudo-parallel corpus using initial source to target model (Forward translated corpus)*




*Target side of pseudo-parallel corpus is noisy*
- *Train the S-T mode on pseudo-parallel corpora*
- *Tune on true parallel corpora*




***Why is self-training useful?***
- *Adaptation to source language domain*
- *Prevent overfitting by exposure to diverse corpora*

*Works well if the initial model is reasonably good*

# How do we evaluate MT systems?

# Evaluation of MT output

- How do we judge a good translation?
- Can a machine do this?
- Why should a machine do this?
  - Because human evaluation is time-consuming and expensive!
  - Not suitable for rapid iteration of feature improvements

# Human Evaluation

## Direct Assessment

**How do you rate your Olympic experience?**

— Reference

**How do you value the Olympic experience?**

— Candidate translation

**Adequacy:**

Is the meaning translated correctly?

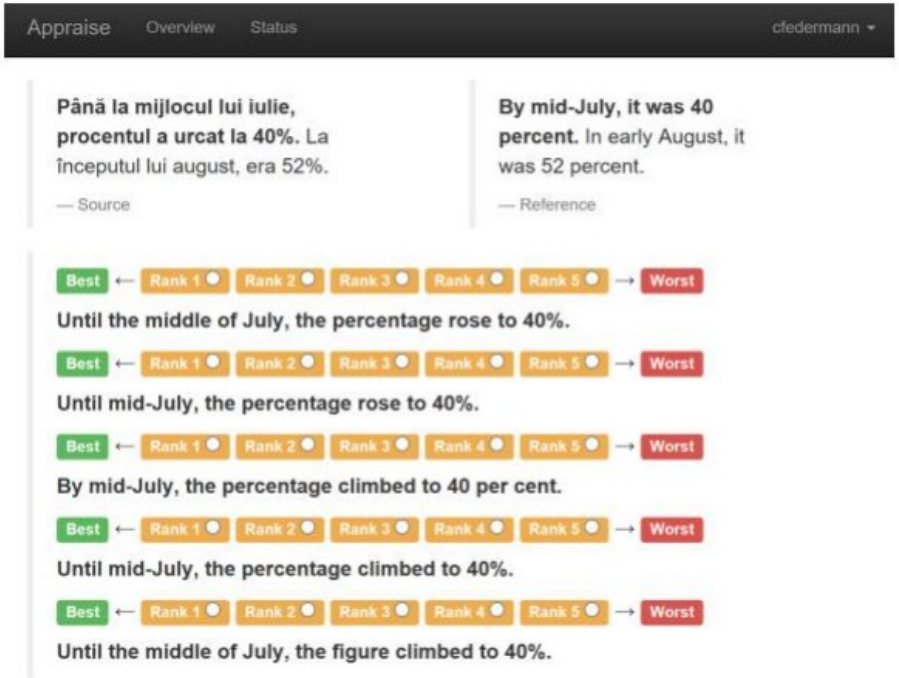
5 = All
4 = Most
3 = Much
2 = Little
1 = None

**Fluency:**

Is the sentence grammatically valid?

5 = Flawless
4 = Good
3 = Non-native
2 = Disfluent
1 = Incomprehensible

## Ranking Translations



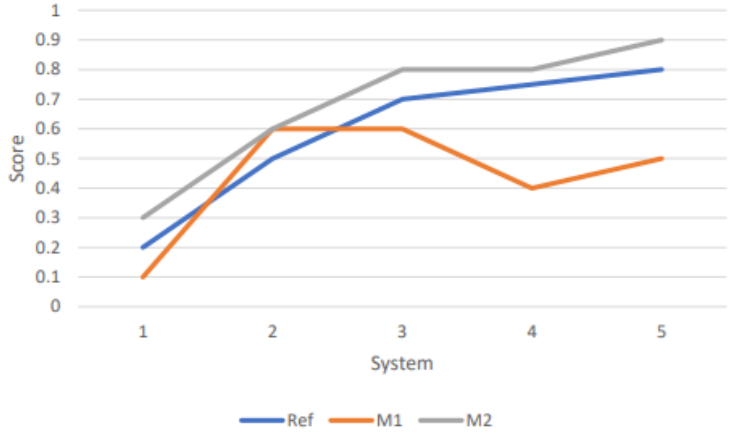

$$\text{score}(S_i) = \frac{1}{|\{S\}|} \sum_{S_j \neq S_i} \frac{\text{wins}(S_i, S_j)}{\text{wins}(S_i, S_j) + \text{wins}(S_j, S_i)}$$

# Some popular automatic evaluation metrics

- BLEU (Bilingual Evaluation Understudy)
- TER (Translation Edit Rate)
- METEOR (Metric for Evaluation of Translation with Explicit Ordering)

How good is an automatic metric?

How well does it correlate with human judgment?



# Learning evaluation metrics

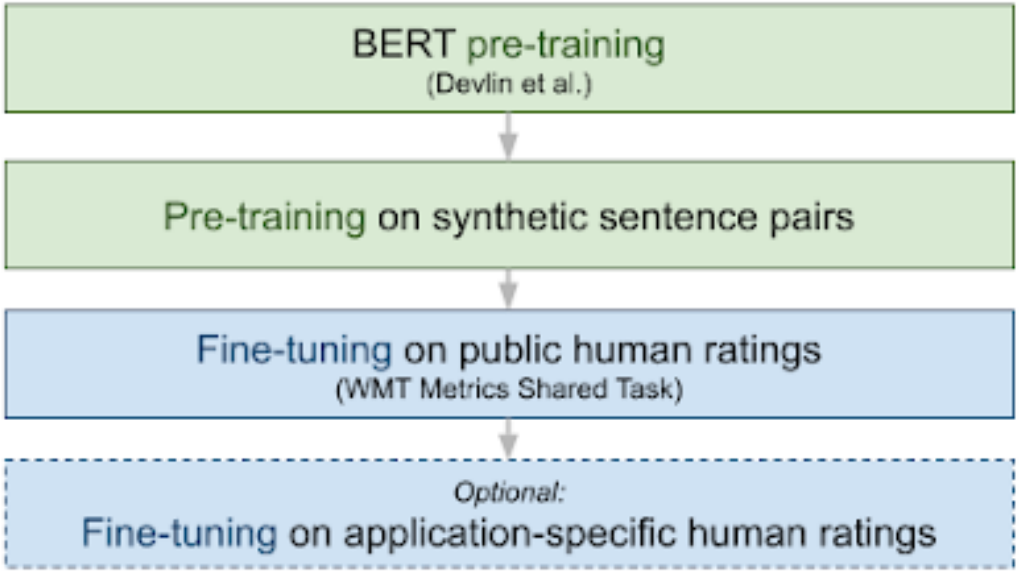
- However, lots of issues exist with metrics like BLEU

- They don’t consider semantic meaning of sentences
- Hence, rephrased outputs are given low scores

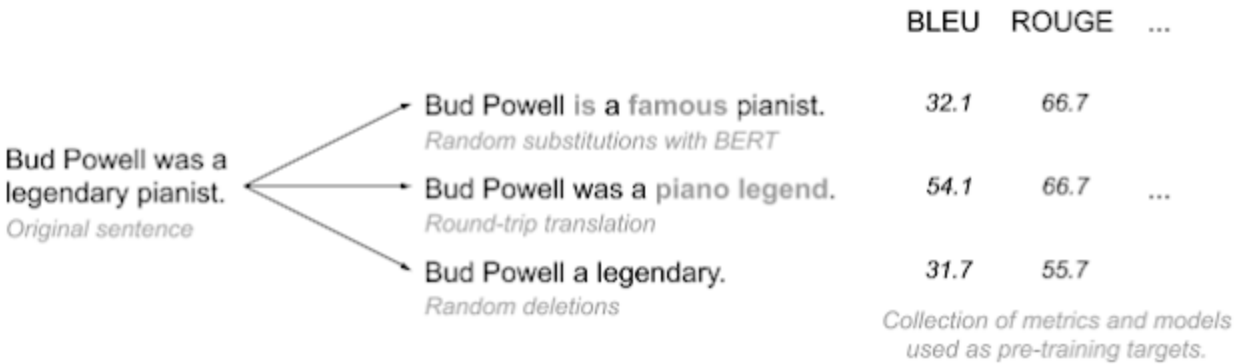
- BLEU diverges with human ratings once systems cross performance threshold

- Solution: Use trained neural models for evaluating NMT systems!

# BLEURT: BLEU-BERT



# BLEURT




BLEURT's data generation process combines random perturbations and scoring with pre-existing metrics and models.

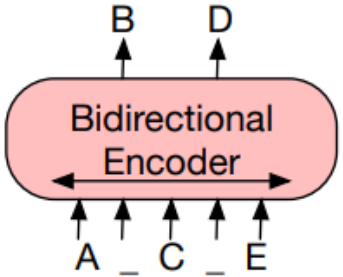
# How to improve performance on low-resource languages?

# Google’s NMT Performance

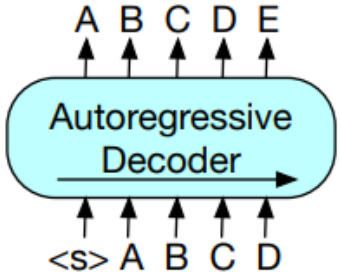
- image1.gif (1000×750) (bp.blogspot.com)

# BART - Pre-training for Seq2Seq tasks

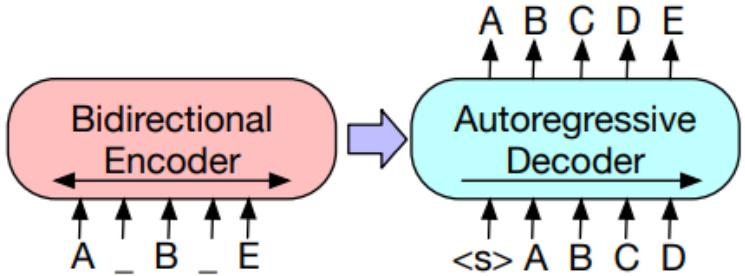
- Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension, Lewis et. al., 2019
- BERT pre-training is useful for Encoder-only settings
- BART introduces pre-training strategy for Seq2Seq models
- Results in SoTA in Seq2Seq tasks like Machine Translation and Summarization.




(a) BERT: Random tokens are replaced with masks, and the document is encoded bidirectionally. Missing tokens are predicted independently, so BERT cannot easily be used for generation.




(b) GPT: Tokens are predicted auto-regressively, meaning GPT can be used for generation. However words can only condition on leftward context, so it cannot learn bidirectional interactions.




(c) BART: Inputs to the encoder need not be aligned with decoder outputs, allowing arbitary noise transformations. Here, a document has been corrupted by replacing spans of text with mask symbols. The corrupted document (left) is encoded with a bidirectional model, and then the likelihood of the original document (right) is calculated with an autoregressive decoder. For fine-tuning, an uncorrupted document is input to both the encoder and decoder, and we use representations from the final hidden state of the decoder.

Figure 1: A schematic comparison of BART with BERT (Devlin et al., 2019) and GPT (Radford et al., 2018).

# Seq2Seq Pre-training tasks

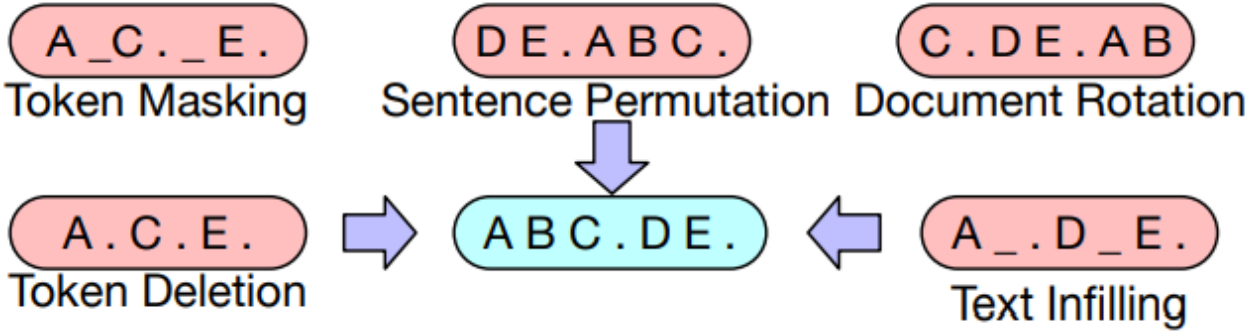



Figure 2: Transformations for noising the input that we experiment with. These transformations can be composed.

# mBART: Multi-Lingual BART

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

- Repeat BART pre-training on CC25 corpus
- Monolingual corpus of 25 languages
- A subset of *Common Crawl*
- A crawl of the internet

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

Table 1: **Languages and Statistics of the CC25 Corpus.** A list of 25 languages ranked with monolingual corpus size. Throughout this paper, we replace the language names with their ISO codes for simplicity. (*) Chinese and Japanese corpus are not segmented, so the tokens counts here are sentences counts

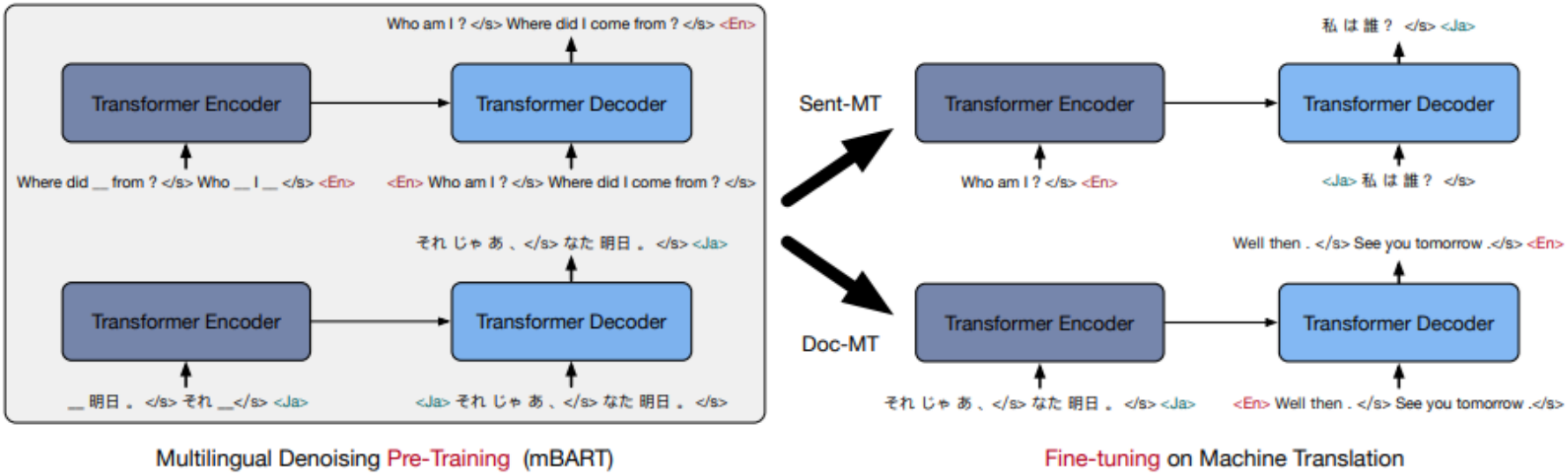



Figure 1: Framework for our Multilingual Denoising Pre-training (left) and fine-tuning on downstream MT tasks (right), where we use (1) sentence permutation (2) word-span masking as the injected noise. A special language id token is added at both the encoder and decoder. One multilingual pre-trained model is used for all tasks.

| 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).

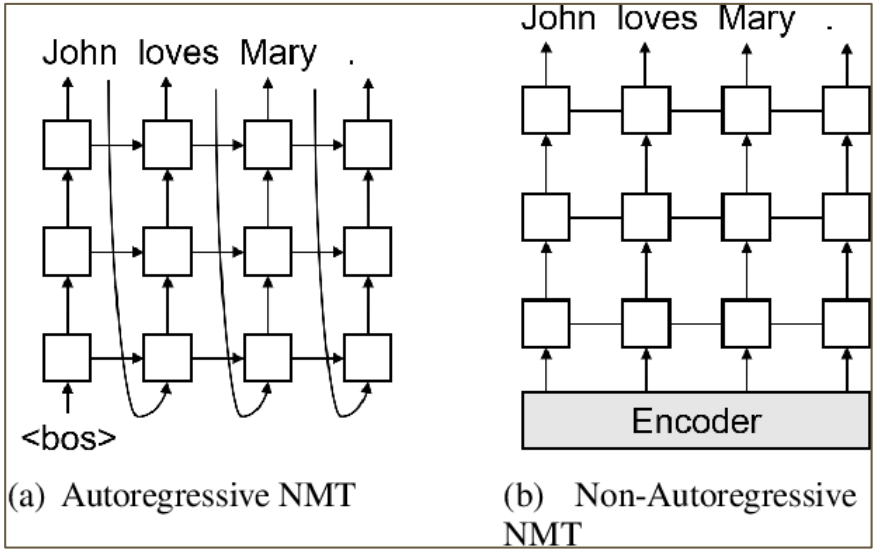
# How to make MT systems faster?

- What is the bottleneck in current MT systems?

- What is the bottleneck in current MT systems?
- Is encoder or decoder part of the architecture the bottleneck?

# Autoregressive vs Non-Autoregressive

- No *sequential* nature in non-autoregressive
- Trade-off of *speed* vs *accuracy*




arxiv:1906.02041

(Machine Translation terminology)

# Non Autoregressive Translation (NAT)

- Need to predict the length of output
- Output may not be grammatical
- Lack of explicit conditioning
- Can introduce additional refinement layers

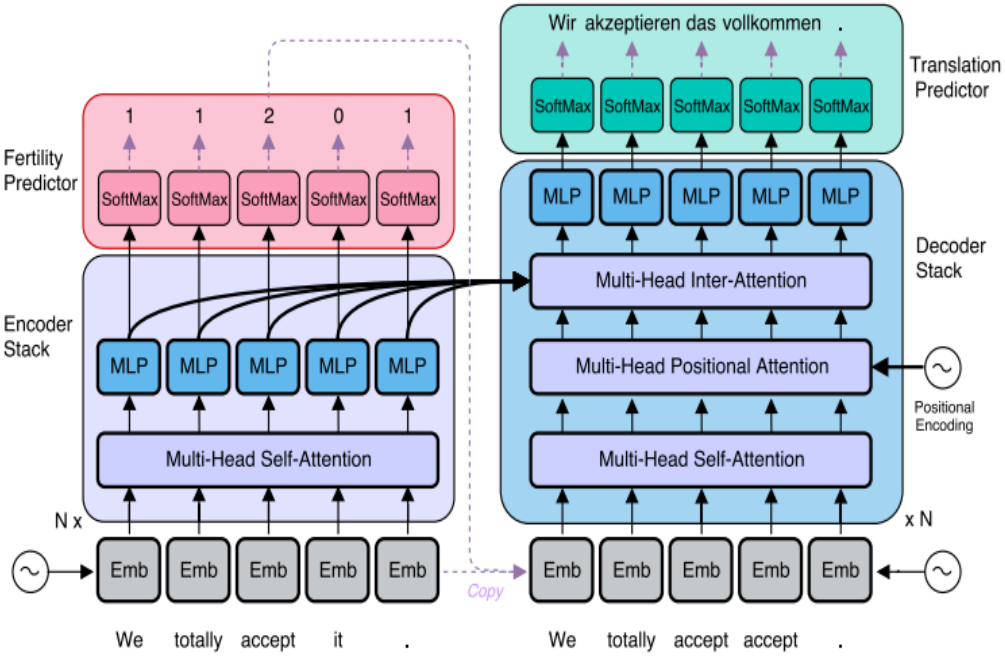



Figure 2: The architecture of the NAT, where the black solid arrows represent differentiable connections and the purple dashed arrows are non-differentiable operations. Each sublayer inside the encoder and decoder stacks also includes layer normalization and a residual connection.

# Felix for Non-Autoregressive generation

- Non-Autoregressive models can be used for any generation task!
- Felix shows it on Seq2Seq tasks
- Pose generation as a pipeline:
- Tagging+reordering+filling
- Tag words to keep, delete, or insert
- Reorder using POINTER network
- Fill in missing words using MLM

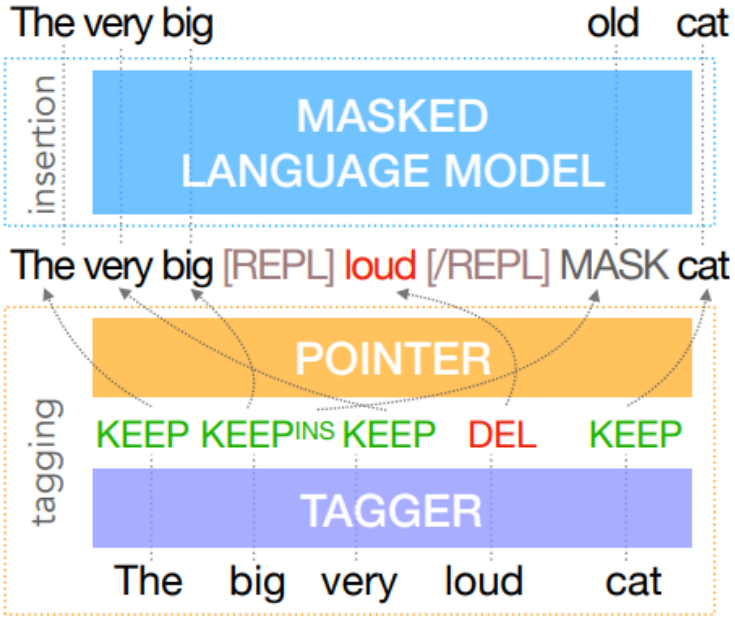



Figure 1: FELIX transforms the source *"The big very loud cat"* into the target text *"The very big old cat"*.