# Deep Dive into NMT

## with NMT Start-up Work Experience

Choi Minjoo (All for One)

# Choi Minjoo
## (Judy Choi)

- Careers
  - 2012 ~ 2019 **Malware Analyst**
  - 2018 **Working Holiday** in France
  - 2020 ~ **M.S** in Kangwon Univ
    - Intelligence Software Lab
      - NLP (Machine Translation)
  - 2021 ~ 2022 **Bering Lab (NLP Researcher & Engineer)**

- SNS
  - https://www.facebook.com/minjoo.choi.562/
  - https://github.com/Judy-Choi

# Contents


1. Instruction
2. Dataset
3. Model
4. Evaluation
5. Appendix


# 1. Instruction

# 1. Instruction

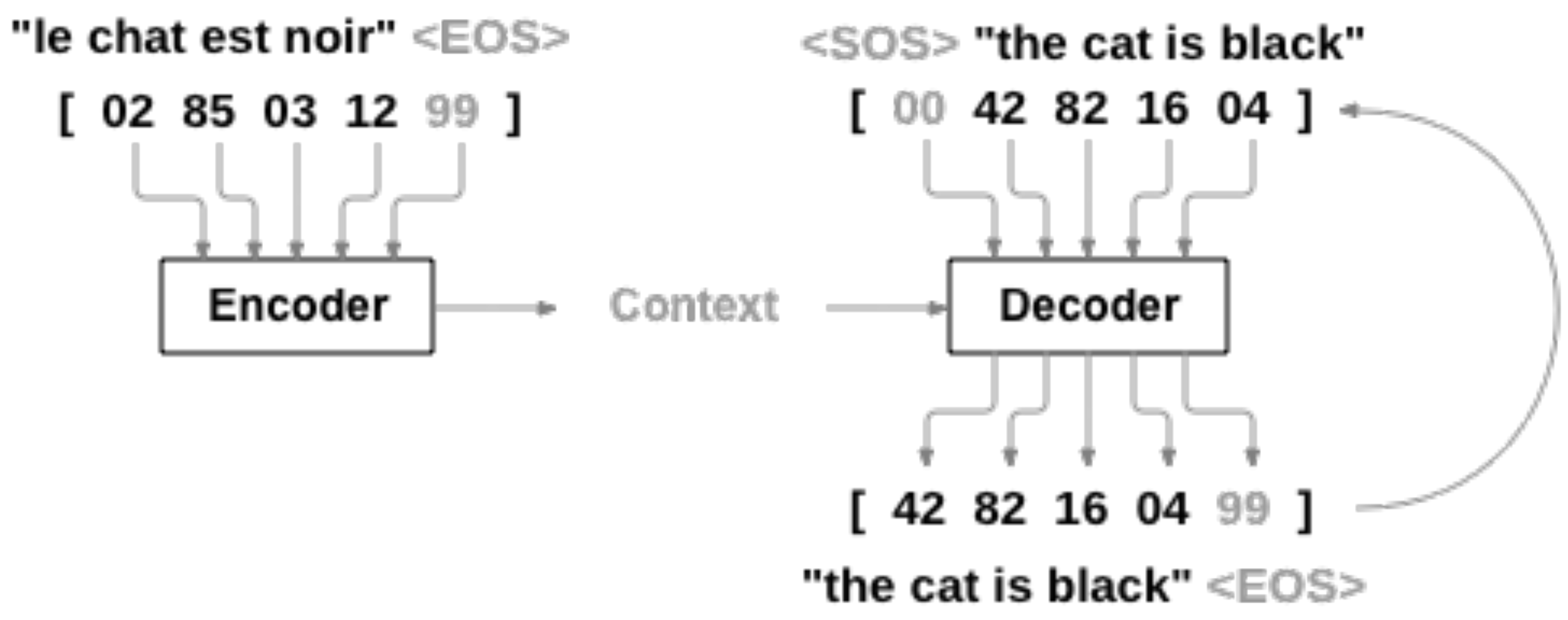
## Machine Translation

- Process when a computer software translates text from one language to another without human involvement

# 1. Instruction

## Neural Machine Translation (NMT)

- Train translation model with language sentence pair
  - Dataset feature : [Source sentence] - [Target sentence]
    - Don’t need to build phrase analyzer / dictionary
    - Translate sentence by sentence -> Translate in context is possible

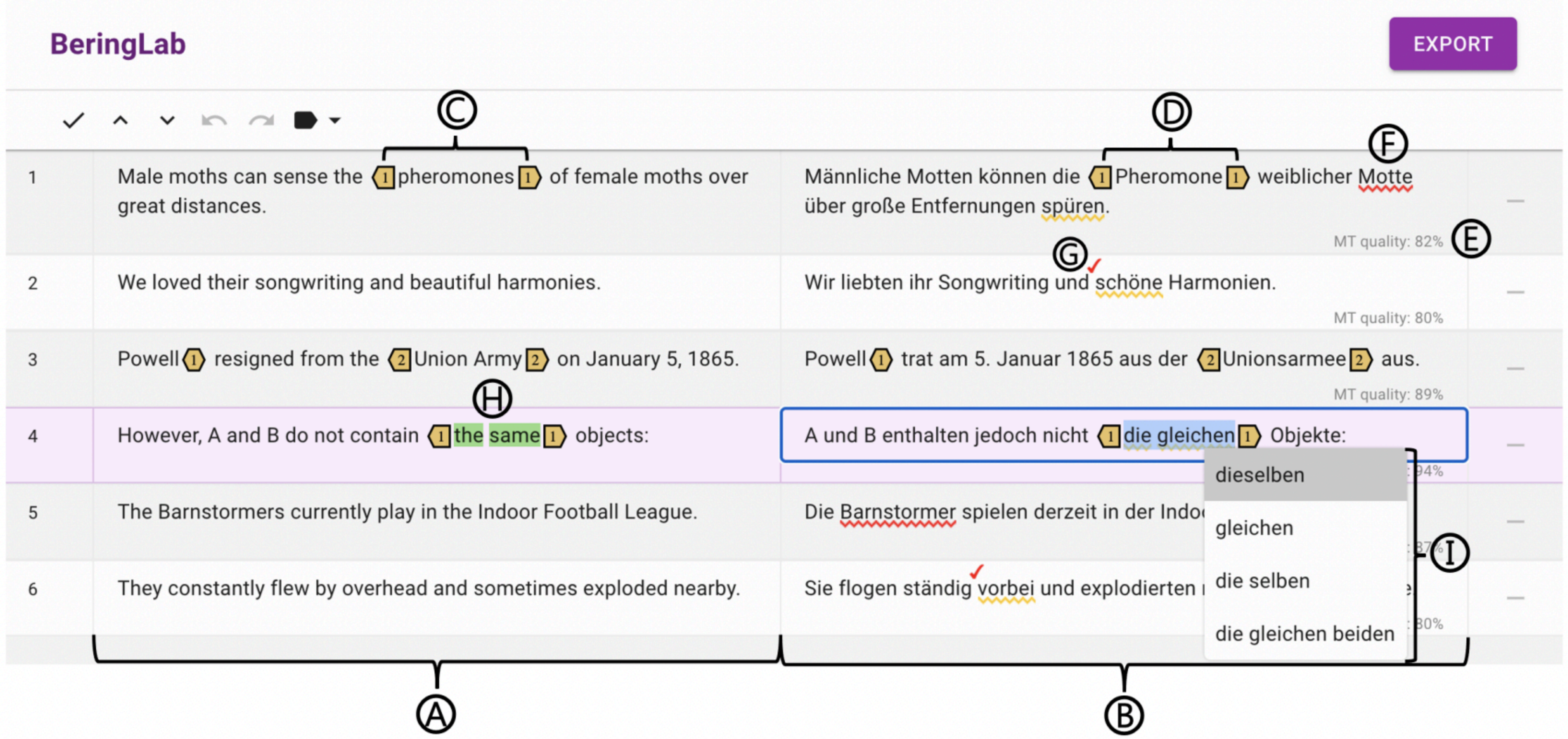



https://tutorials.pytorch.kr/_images/seq2seq.png

# 1. Instruction

## NMT and more

- Translation Task
  - Domain-Specific (Biomedical, News, Patent, Chat…)
- Evaluation Task
  - QE (Quality Estimation)
- Other Task
  - APE (Automatic Post-Editing)
  - Translation Suggestion




https://arxiv.org/pdf/2105.12172.pdf

# Set your Goal

# 2. Dataset

# 2. Dataset
## Open Dataset

- Popular Corpus
  - OPUS
    - Collection of translated texts from the web
  - TED, OpenSubTitles (Colloquial Style)
  - Europarl (News)
  - FLORES (Wikipedia)

- Competition
  - WMT
  - Kaggle

- * In Korea
  - AI Hub
    - https://www.aihub.or.kr/

# 2. Dataset
## Crawling

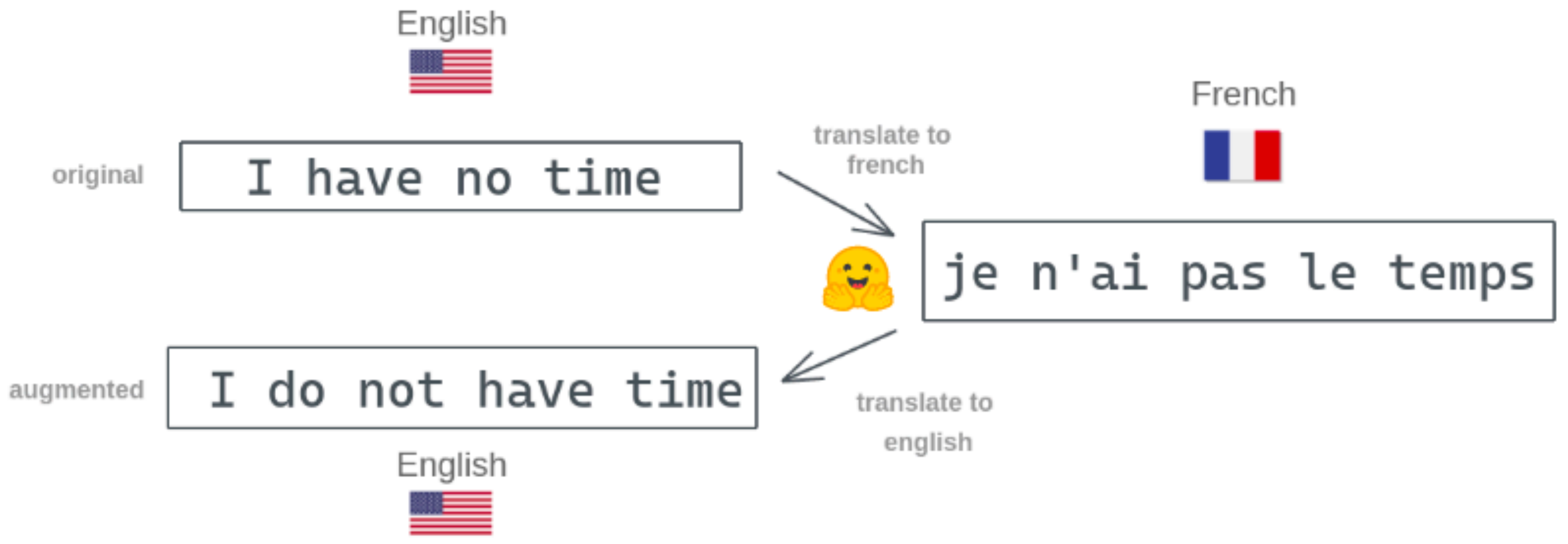
- Crawling website
  - Google Patent
  - SNS, News, Google Trend, Stock..
- Crawling Library
  - BeautifulSoup
  - Selenium
  - Scrapy

# 2. Dataset

## Generation

- By Human
  - Professional
    (Domain-Specific Task)
  - Human essential Task (ex : APE)
  - Chatbot
  - Scenario (Generation Task)

- Artificial Generation
  - Back-Translation




https://amitness.com/back-translation/

# 2. Dataset

## Example : Legal NMT

- Translated by human
  - Legal Article
  - Contract Document
- Dictionary
  - Legal words
  - Company name (NER)
- News Article (by Crawled Open Dataset)

# 2. Dataset
## Filtering

- Rule
  - Filter by rule
    - Length of sentence
    - Remove blank, special character..
  - Fast, Simple, Clear
  - Perl script

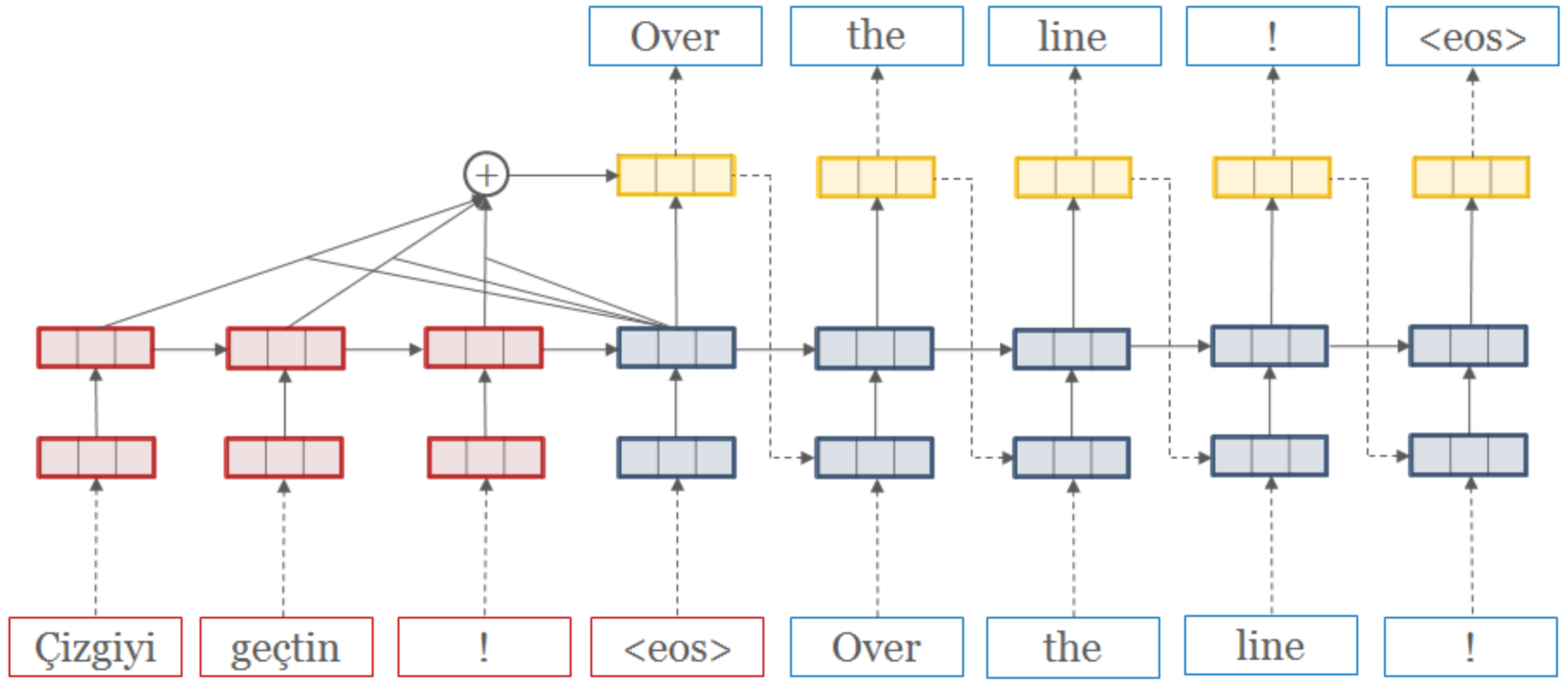
- Embedding & Similarity
  1. Filter by Deep Learning
     - Multilingual, Semantic
     - Open source
       - Universal Encoder (A little old method)
       - LASER (by Facebook AI)
  2. Calculate Similarity
     - Cosine Similarity
     - FAISS

# 3. Model

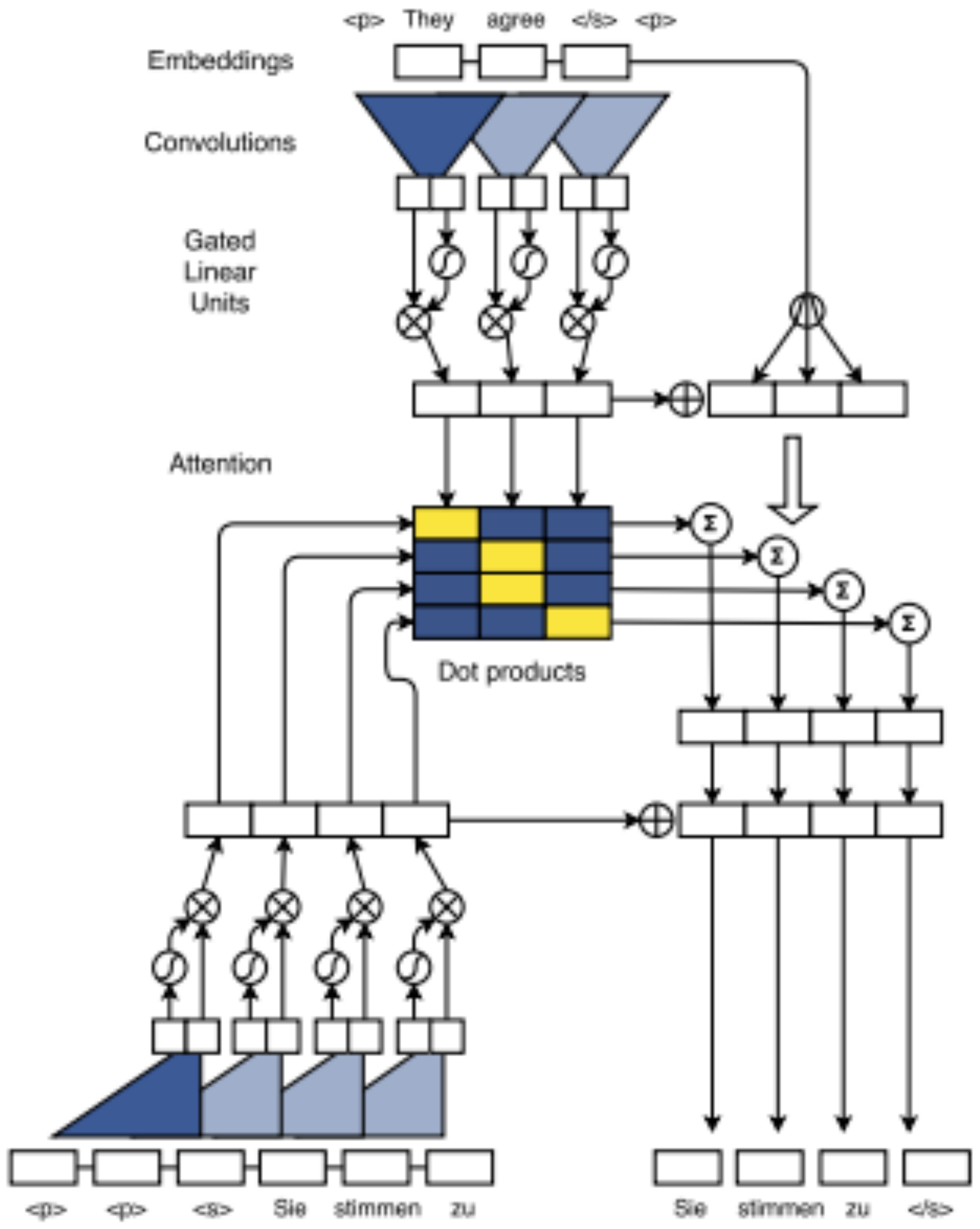
# 3. Model

## Open Source

- OpenNMT
  - Havard NLP group & Systran
  - TensorFlow, PyTorch




- FairSeq
  - Meta (Facebook AI)
  - PyTorch

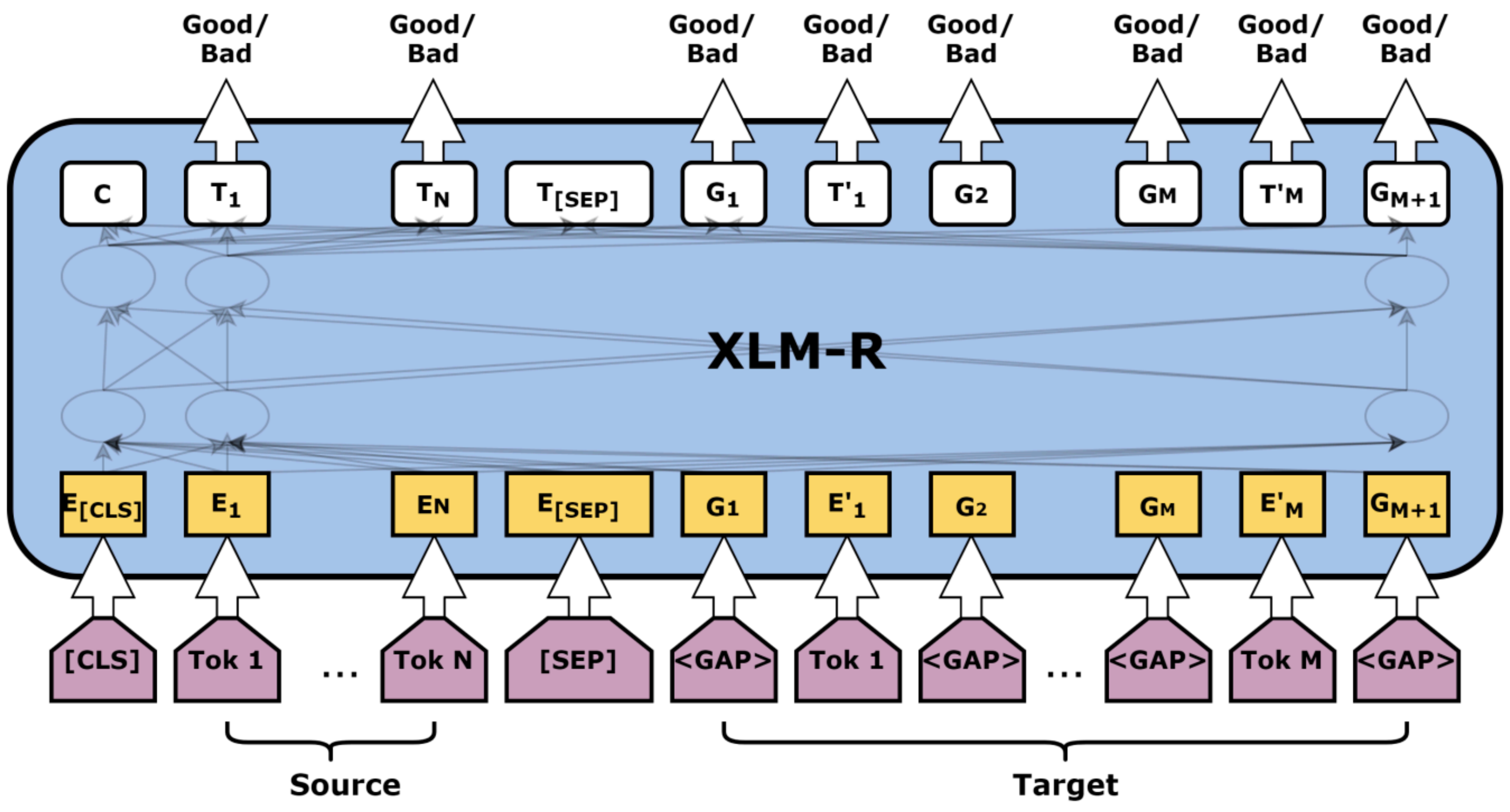



**F**acebook **AI** **R**esearch **Seq**uence-to-Sequence

# 3. Model

## Open Source

- XLM-RoBERTa
  - Cross-Lingual Model of RoBERTa
  - QE, APE




https://aclanthology.org/2021.acl-short.55.pdf

# 3. Model

## Fine-tune

- Find your own profit hyper-parameter!

config.yml

```yaml
# General opts
save_model: en-fr
save_checkpoint_steps: 10000
valid_steps: 1000
train_steps: 200000

# Batching
queue_size: 10000
bucket_size: 32768
world_size: 1
gpu_ranks: -0
batch_type: "tokens"
batch_size: 4096
valid_batch_size: 16
max_generator_batches: 2
accum_count: [2]
accum_steps: [0]
```

```yaml
# Optimization
model_dtype: "fp32"
optim: "adam"
learning_rate: 2
warmup_steps: 8000
decay_method: "noam"
adam_beta2: 0.998
max_grad_norm: 0
label_smoothing: 0.1
param_init: 0
param_init_glorot: true
normalization: "tokens"
```

```yaml
# Model
encoder_type: transformer
decoder_type: transformer
position_encoding: true
enc_layers: 6
dec_layers: 6
heads: 8
rnn_size: 512
word_vec_size: 512
transformer_ff: 2048
dropout_steps: [0]
dropout: [0.1]
attention_dropout: [0.1]
```

- Trade-off
  - Speed or Performance

- Resource Capacity
  - Have enough memory?

# 4. Evaluation

# 4. Evaluation

## Evaluation Metric

- BLEU score (Bilingual Evaluation Understudy Score)
  - Evaluation method by N-gram

$$p_n = \frac{\sum_{C \in \{Candidates\}} \sum_{n\text{-}gram \in C} Count_{clip}(n\text{-}gram)}{\sum_{C' \in \{Candidates\}} \sum_{n\text{-}gram' \in C'} Count(n\text{-}gram')}.$$

```python
# 1 word different
from nltk.translate.bleu_score import sentence_bleu
reference = [['the', 'quick', 'brown', 'fox', 'jumped', 'over', 'the', 'lazy', 'dog']]
candidate = ['the', 'fast', 'brown', 'fox', 'jumped', 'over', 'the', 'lazy', 'dog']
score = sentence_bleu(reference, candidate)
print("{:.2f}".format(score))
```

0.75

```python
# 2 word different
from nltk.translate.bleu_score import sentence_bleu
reference = [['the', 'quick', 'brown', 'fox', 'jumped', 'over', 'the', 'lazy', 'dog']]
candidate = ['the', 'fast', 'brown', 'fox', 'jumped', 'over', 'the', 'sleepy', 'dog']
score = sentence_bleu(reference, candidate)
print("{:.2f}".format(score))
```

0.49

  - Moses multi-bleu perl
    - https://github.com/moses-smt/mosesdecoder/blob/master/scripts/generic/multi-bleu.perl

https://machinelearningmastery.com/calculate-bleu-score-for-text-python/

# 4. Evaluation

## Evaluation Metric

- MAE
  - Mean absolute error
  - QE task

$$\frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$$

- RMSE
  - Root mean squared error
  - QE task

$$\sqrt{\sum_{i=1}^{n}\frac{(\hat{y}_i - y_i)^2}{n}}$$

- TER (HTER)
  - **T**ranslation **E**rror **R**ate
  - APE, QE task

https://www.codingprof.com/3-ways-to-calculate-the-mean-absolute-error-mae-in-r-examples/
https://mobile.twitter.com/PrasoonPratham/status/1393144574638530561/photo/1
https://github.com/jhclark/tercom

# 4. Evaluation

## Human Evaluation

- How can evaluate well by **human**?
  - Human can make mistake
  - How can we catch human mistake?
  - How can we make human evaluate faster and more accurately?

- How can evaluate measured result by **BLUE**?
  - Metric can calculate incorrect score
  - How can we catch incorrect score?

# Now we can build NMT!

And let's think about more…

# 5. Appendix

# 5. Appendix

## Let’s release our NMT service!

- Back-end
  - Server
    - AWS, Docker, Django..
    - Tokenizer & NMT model
  - Store informations
    - query, account…
- Front-end
  - Web site or Application
  - Get query and send to Server

# 5. Appendix

## Researcher != Engineer

- Researcher
  - Improve existing problems
  - '**Deep**' insight
    - Model, Formula
- Engineer
  - Use existing technology
  - '**Wide**' insight
    - Speed, Capacity
  - Business insight is needed

# 5. Appendix

Let’s step up

- WMT
  - **W**orkshop on **M**achine **T**ranslation
  - EMNLP conference on Machine Translation
  - https://statmt.org/wmt22/

Now we are the NMT specialist!

# Q&A