

Curriculum Learning

An Efficient Learning Paradigm

ICARC Tutorial Session 2026

Curriculum Learning in NLP

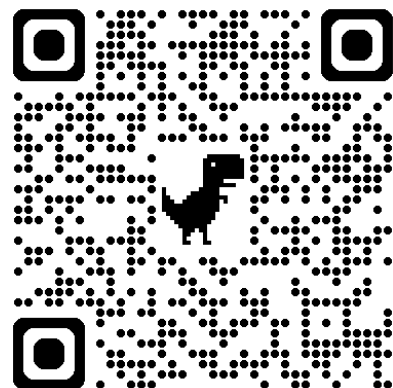


Menan Velayuthan.

Researcher & PhD Candidate | 

```
terminal
$ pwd
/PhD/The_Netherlands/Utrecht_University/NLP_and_Society_Lab/Data_Divers
```

I am a PhD candidate at the [NLP & Society Lab](#), Utrecht University, working on the [DataDivers](#) project under the supervision of Professor [Dong Nguyen](#). My research focuses on data diversity for robust and fair language models, with an emphasis on inclusive and egalitarian AI.



PyTorchSL Hands-on

Welcome to the PyTorchSL learning series.
This repository contains hands-on lessons designed for beginners to learn PyTorch step by step.

Spark Track (Beginner) ✨

Lets start by sparking the flaming!

PyTorchSL Hands-on ✨

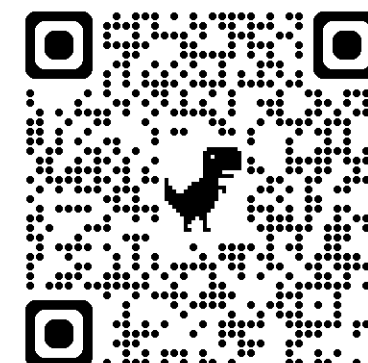
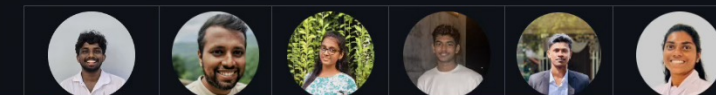
Welcome to the PyTorchSL Spark Track ✨ — a beginner-friendly hands-on journey into PyTorch.

Below is the lesson schedule for the Spark Track.

Lesson	Date	Notebook Name	Colab	Recording
lesson1	22-11-2025	lesson0	Open in Colab	Watch Recording
lesson2	6-12-2025	lesson1	Open in Colab	Watch Recording
lesson3	20-12-2025	1. Non-linearity and deep networks 2. Vision first steps	Open in Colab Open in Colab	Watch Recording

More lessons will be added as the Spark Track continues.
Stay tuned!

👥 Meet the Team



What to Expect?

- Quick Intro to Natural Language Processing (NLP).
- NLP Tasks
- Discussing 2 works using Curriculum Learning in NLP.
[Zaremba and Ilya \(2015\)](#)
[Ranathunga et al. \(2024\)](#)

What I want you to take from this talk?

- Try to connect the task the researchers try to solve, and the difficulty function they use.
- Understand that each problem demands its own conceptualization of the curriculum.
- Observe the justification of using the curriculum learning paradigm in the works we discuss.

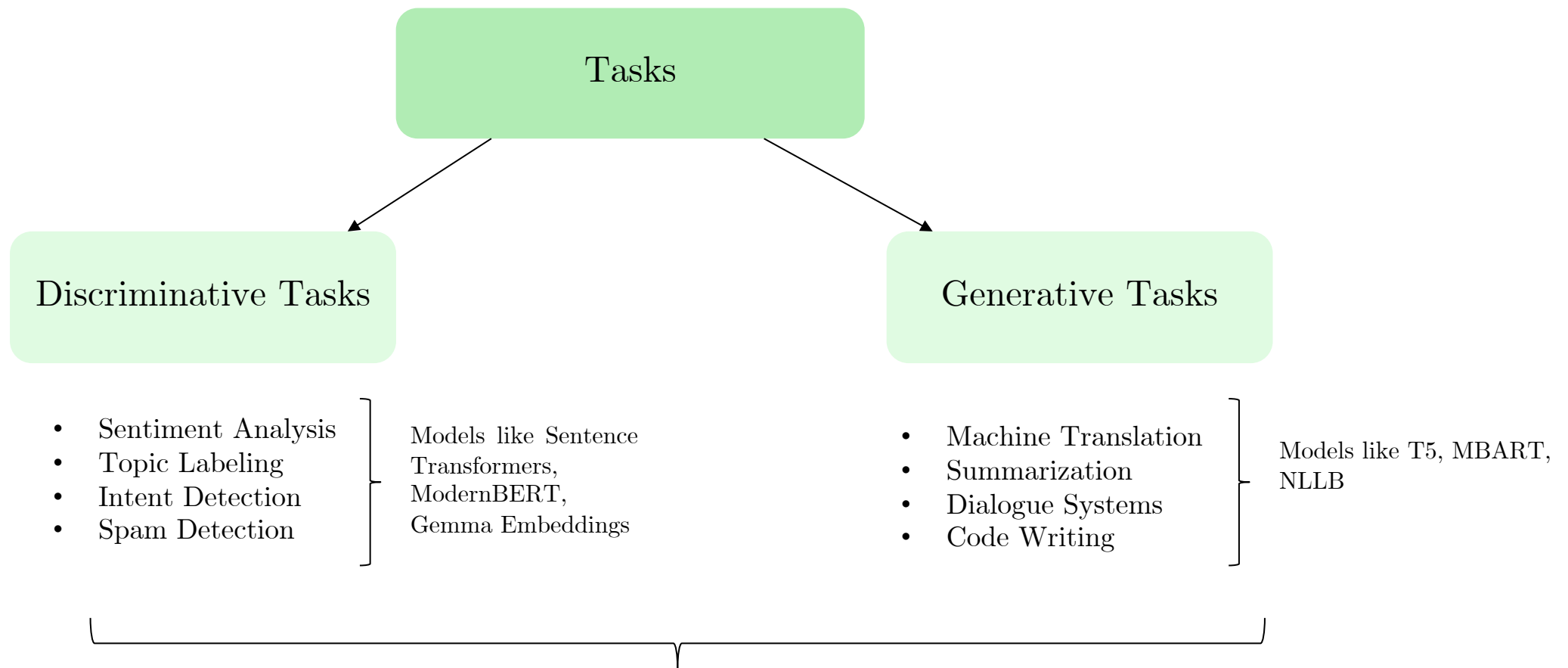
“Natural language processing (NLP) is technology that allows computers to interpret, manipulate, and comprehend **human language**.”

Source: <https://aws.amazon.com/what-is/nlp/>

Human Language Modalities

Textual Data

Speech Data



LEARNING TO EXECUTE

Wojciech Zaremba*
New York University
woj.zaremba@gmail.com

Ilya Sutskever
Google
ilyasu@google.com

ABSTRACT

Recurrent Neural Networks (RNNs) with Long Short-Term Memory units (LSTM) are widely used because they are expressive and are easy to train. Our interest lies in empirically evaluating the expressiveness and the learnability of LSTMs in the sequence-to-sequence regime by training them to evaluate short computer programs, a domain that has traditionally been seen as too complex for neural networks. We consider a simple class of programs that can be evaluated with a single left-to-right pass using constant memory. Our main result is that LSTMs can learn to map the character-level representations of such programs to their correct outputs. Notably, it was necessary to use curriculum learning, and while conventional curriculum learning proved ineffective, we developed a new variant of curriculum learning that improved our networks' performance in all experimental conditions. The improved curriculum had a dramatic impact on an addition problem, making it possible to train an LSTM to add two 9-digit numbers with 99% accuracy.

Program Evaluation

Input:

```
j=8584
for x in range(8):
    j+=920
b=(1500+j)
print ( (b+7567) )
```

Target: 25011.

Addition

Input:

```
print (398345+425098)
```

Target: 823443

Memorization

Input

2345

Target: 2345

Program Evaluation

Length of Digits

Nesting

```
i = 42
print((i + 89))
```

```
i = 8827
c = (i - 5347)
print( ((c + 8704) if 2641 < 8500 else 5308) ).
```

```
j = 611989
for x in range(4):
    j += 7638
b = (j - 150000)
print( (b * 3) ).
```

Addition

Length of Digits

Length = 3
`print(123+456)`

Length = 6
`print(398345+425098)`

Length = 9
`print(123456789+987654321)`

Memorization

Length of Digits

Input: 12345
Output: 12345

Input: 54321
Output: 12345

Input: 54321; 54321
Output: 12345

Training Strategies

- **No curriculum (baseline):** The usual standard way of training on the available data. In this case, it is the final version of the task.
- **Naïve curriculum strategy (naïve):** The standard curriculum we saw, we go from the easiest data to the hardest, in a gradual incremental way.
- **Mixed strategy (mix):** This is a random approach, when selecting a training sample, we randomly pick the sample based on its difficulty.
- **Combined strategy:** A combination of the naïve and mix approach. Half the time the model selects data in the naïve gradually increasing difficulty manner, other half it goes with random difficulty selection method.

From there experiments, it is seen that “Combined strategy” outperformed all the other strategy.

Hidden State Allocation Hypothesis

“....based the on empirical results, the naive strategy of curriculum learning can sometimes be worse than learning with the target distribution.”

Why?

“Indeed, the network has no incentive to utilize only a fraction of its state, and it is always better to make use of its entire memory capacity. This implies that the harder examples would require a restructuring of its memory patterns.”

Quality Does Matter: A Detailed Look at the Quality and Utility of Web-Mined Parallel Corpora

**Surangika Ranathunga¹, Nisansa de Silva², Menan Velayuthan²,
Aloka Fernando², Charitha Rathnayake²**

¹Massey University, Palmerston North, New Zealand, 4443

²Dept. of Computer Science & Engineering, University of Moratuwa, 10400, Sri Lanka
s.ranathunga@massey.ac.nz
{NisansaDdS, velayuthan.22, alokaf, charitha.18}@cse.mrt.ac.lk

Best Low Resource Paper Award at EACL 2024

Select source

Pick source first

Contribute

Publications

Corpora

Synthetic

Dashboard

Resources for English (en) – Sinhala (si) (33 found)

Search by corpus

Type a corpus name...

Rows

10

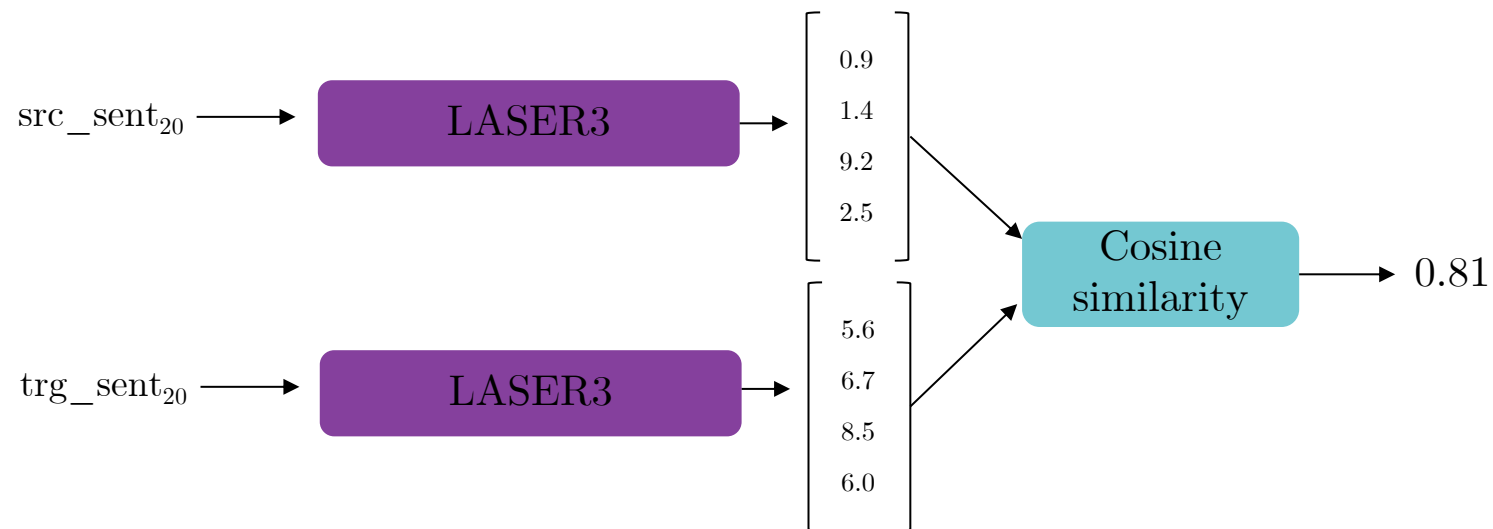
Next

1–10 / 33

CORPUS	<div> <div>↓</div> <div>↑</div> </div> <div> <div>↓</div> <div>↑</div> </div>	<div> <div>↓</div> <div>↑</div> </div> <div> <div>↓</div> <div>↑</div> </div>	SENTENCES	EN TOK	SI TOK	SAMPLE	BILINGUAL	MONOLINGUAL
NLLB v1			24,336,367	229,897,079	227,072,146		<div>moses</div> <div>↓</div> <div>↻</div>	<div>txt en</div> <div>↓</div> <div>↻</div>
CCMatrix v1			6,270,800	47,797,892	43,750,554		<div>moses</div> <div>↓</div> <div>↻</div>	<div>txt en</div> <div>↓</div> <div>↻</div>
OpenSubtitles v2024			3,201,159	18,950,899	16,633,389		<div>moses</div> <div>↓</div> <div>↻</div>	<div>txt en</div> <div>↓</div> <div>↻</div>
XLEnt v1.1			690,187	1,737,128	1,961,398		<div>moses</div> <div>↓</div> <div>↻</div>	<div>txt en</div> <div>↓</div> <div>↻</div>
XLEnt v1			690,187	1,737,128	1,961,398		<div>moses</div> <div>↓</div> <div>↻</div>	<div>txt en</div> <div>↓</div> <div>↻</div>
XLEnt v1.2			690,187	1,737,130	1,961,576		<div>moses</div> <div>↓</div> <div>↻</div>	<div>txt en</div> <div>↓</div> <div>↻</div>
CCAligned v1			619,730	8,672,884	8,327,097		<div>moses</div> <div>↓</div> <div>↻</div>	<div>txt en</div> <div>↓</div> <div>↻</div>
MultiCCAligned v1			619,730	8,673,002	8,327,213		<div>moses</div> <div>↓</div> <div>↻</div>	<div>txt en</div> <div>↓</div> <div>↻</div>
OpenSubtitles v2018			601,164	3,594,769	3,123,232		<div>moses</div> <div>↓</div> <div>↻</div>	<div>txt en</div> <div>↓</div> <div>↻</div>
OpenSubtitles v2016			392,368	2,365,645	2,045,914		<div>moses</div> <div>↓</div> <div>↻</div>	<div>txt en</div> <div>↓</div> <div>↻</div>

Source: <https://opus.nlpl.eu/corpora-search/en&si>

Src (eng)	Trg (si)
src_sent ₁	trg_sent ₁
....
src_sent _N	trg_sent _N



Src (eng)	Trg (si)	Cos sim
src_sent ₁	trg_sent ₁	0.01
....
src_sent ₁₀₀	trg_sent ₁₀₀	0.98
....
src_sent _N	trg_sent _N	0.75

Sort in descending order

Src (eng)	Trg (si)	Cos sim
src_sent ₁₀₀	trg_sent ₁₀₀	0.98
....
src_sent _N	trg_sent _N	0.75
....
src_sent ₁	trg_sent ₁	0.01

Filter Top
25,000

LASER3: <https://github.com/facebookresearch/LASER>

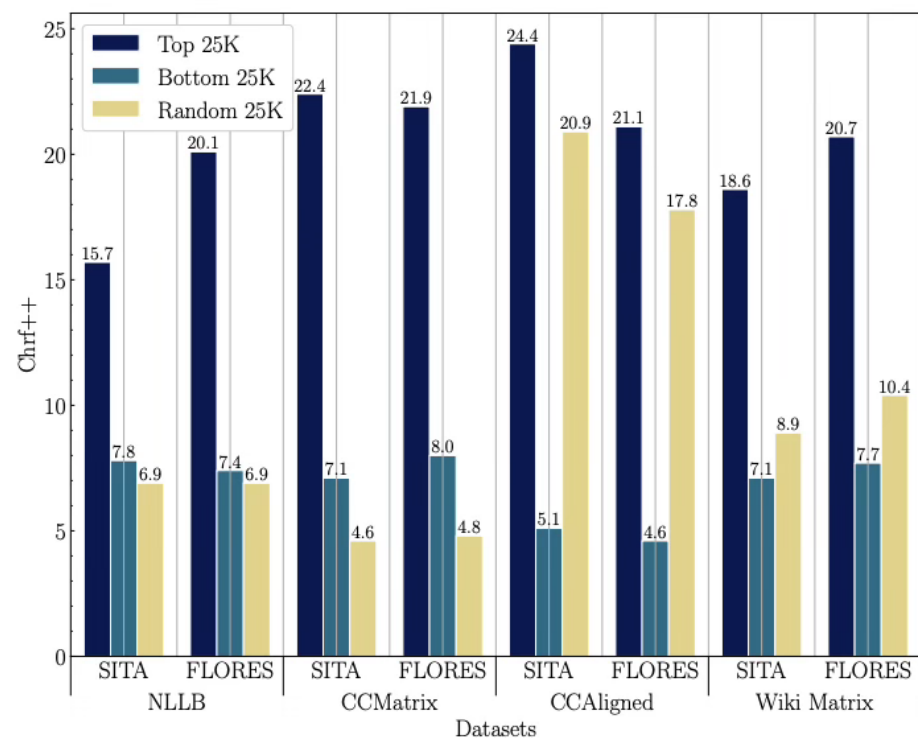


Figure 1: Vanilla-transformer performance trained on Top, Bottom and Random 25K splits of NLLB, CCMatrix, CCAAligned and WikiMatrix for En-Si (higher the better).

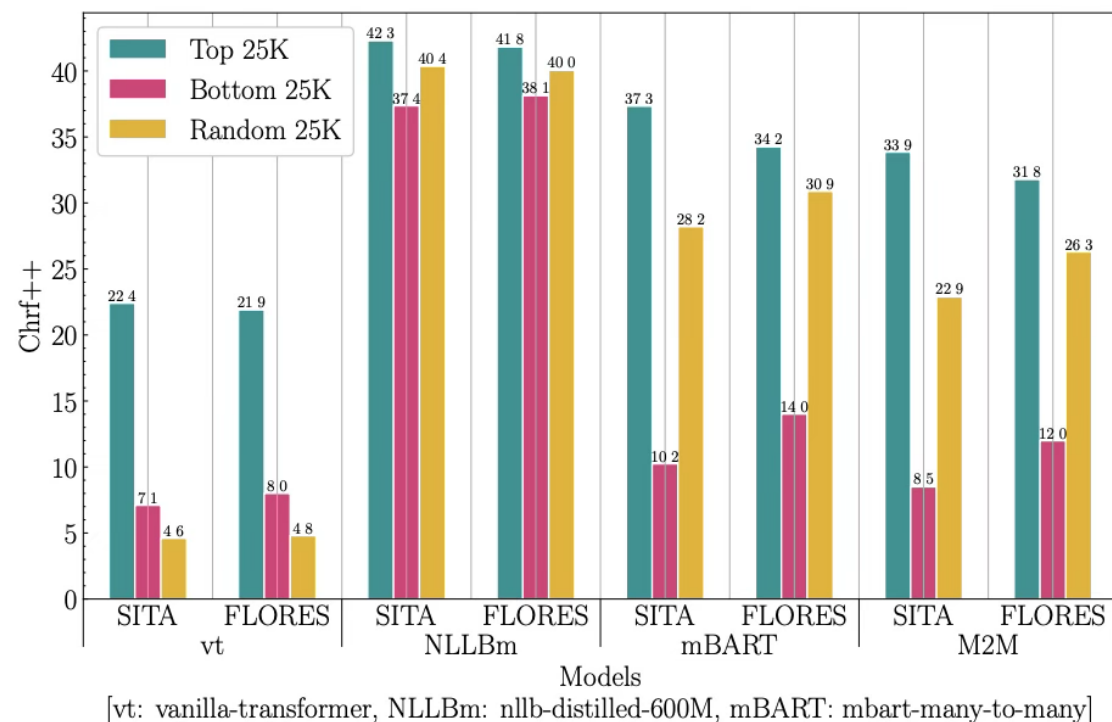


Figure 2: NMT results of different models trained on CCMatrix En-Si top, bottom and average 25K splits.

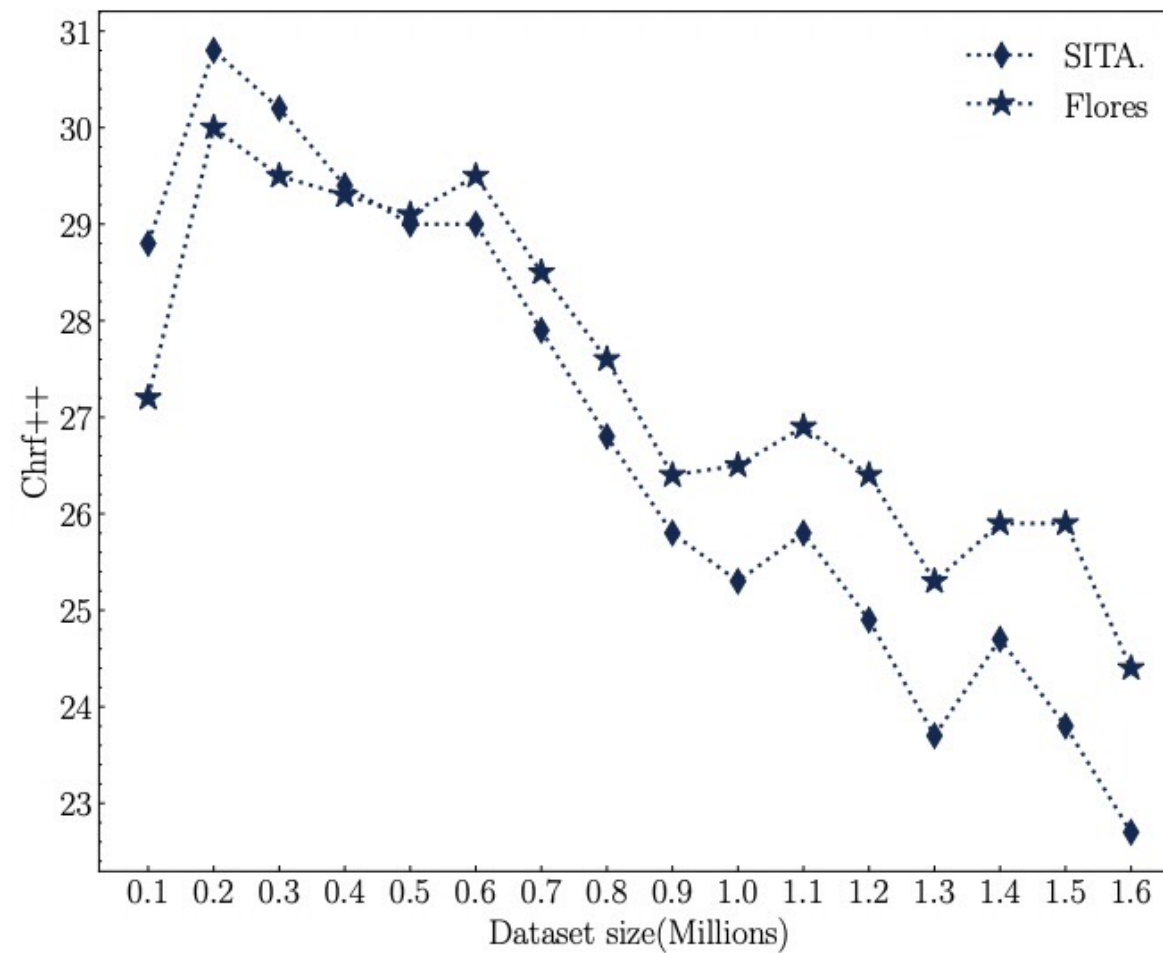


Figure 3: NMT results of vanilla transformer model trained on CCMatrix En-Si in jumps of 100K.

Some practical advice

- Curriculum Learning (CL) is just another learning paradigm, it doesn't promise all the AI problems, sometimes, solutions like simple fine tuning a pretrained model can solve your problem.
- Always work with a baseline (meaning without CL). And see whether CL based approach has some kind of gain when compared the base line.
- These gains doesn't always have to be on performance on a task (for example accuracy of the model). It could be on convergence speed, utilization of less data to achieve same model performance as the entirety of the data.

Thank you!