MUNI
FI

LANGTOK: Language Identification at the Token Level

Emma Bednaříková, FI MUNI

Outline

- Description of the task and the model (dataset, training, evaluation)
- Practical applications
- Optimizing input processing

Outline

- Description of the task and the model (dataset, training, evaluation)
- Practical applications
- Optimizing input processing

LANGTOK model

- model for token classification (demo)
- identification of the language for each token on the input

MUNI FI

Creating the model

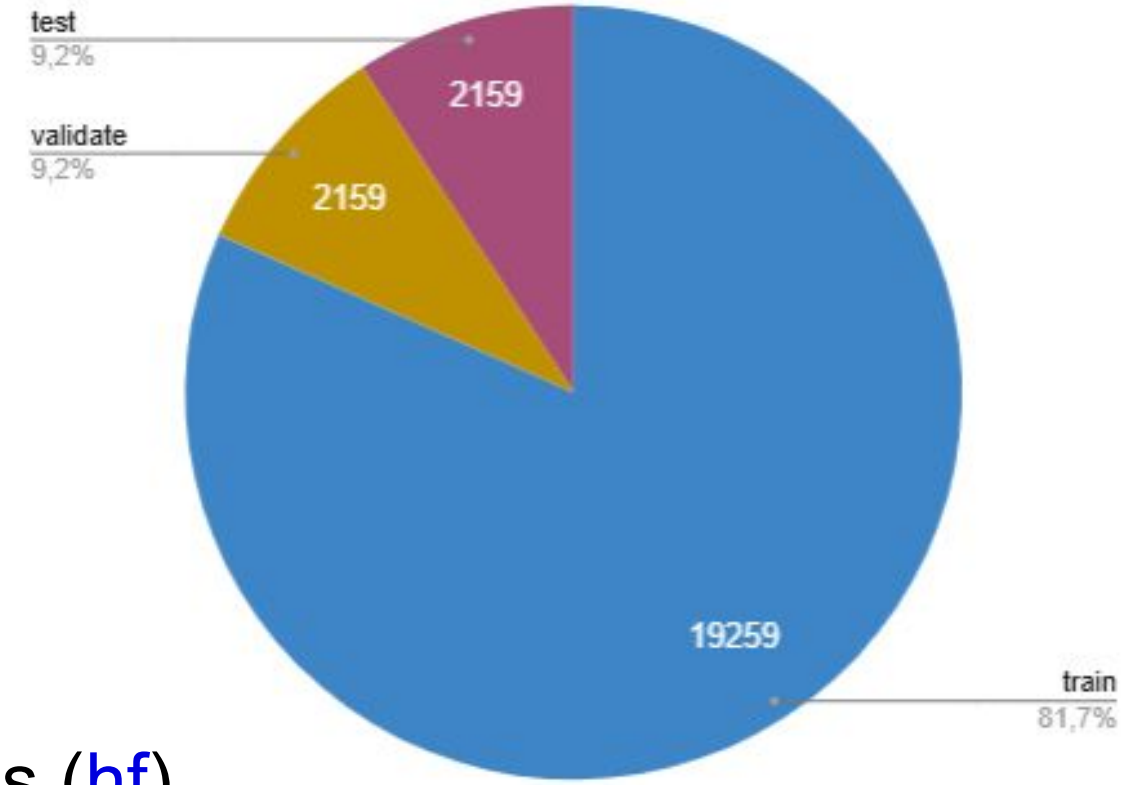
- Token Classification on Hugging Face
- Building a dataset
- Training

Dataset

MUNI FI

Dataset

- raw text from FLORES-200



- **Train**: ca 19 250 modified sentences (hf)
- **Validate**: ca 2 150 modified sentences (hf)
- **Test**: ca 2 150 modified sentences (hf)

MUNI FI

Used Languages

arb	Arabic	ces	Czech
dan	Danish	deu	German
eng	English	fra	French
hat	Haitian Creole	ita	Italian
jpn	Japanese	lin	Lingala
nld	Dutch	pol	Polish
por	Portuguese	rus	Russian
slk	Slovak	spa	Spanish
swe	Swedish	ukr	Ukrainian

MUNI FI

Training

- Pretrained model: google-bert/bert-base-multilingual-cased
- Used parameters from Hugging Face tutorial
  - Batch size: 16
  - Number of training epochs: 3
  - Learning rate: 2e-5
- Duration of the training
  - CPU: cca 7 hours
  - GPU (Tesla T4, apollo): 28 min 28 s

Evaluation

	arb	ces	dan	deu	eng	fra	hat	ita	jpn	lin	nld	pol	por	rus	slk	spa	swe	ukr
arb	6587	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
ces	0	6547	1	7	4	0	2	1	0	2	0	7	4	0	96	0	1	1
dan	0	0	5968	5	0	3	6	3	0	0	10	3	1	0	0	0	40	0
deu	0	0	4	5790	5	3	3	0	0	0	5	1	0	0	1	4	4	0
eng	0	3	3	0	4512	6	1	2	0	1	10	3	2	0	8	1	0	0
fra	0	3	8	0	2	6105	1	3	0	5	6	0	4	0	9	14	0	0
hat	0	2	3	0	1	2	6945	2	0	11	4	6	1	0	4	4	0	0
ita	0	0	0	4	4	17	2	5969	0	3	1	5	7	0	4	20	2	0
jpn	0	0	0	0	5	0	1	0	8345	0	0	0	0	0	0	0	0	0
lin	0	3	3	1	4	0	7	3	0	8034	1	6	2	1	5	3	2	0
nld	0	0	30	8	3	2	4	1	2	3	5877	2	1	0	8	3	5	0
pol	0	5	1	2	2	0	3	1	0	4	1	6904	0	0	16	0	0	0
por	0	2	0	2	9	1	1	2	0	1	10	3	5596	0	6	53	0	0
rus	0	0	0	1	0	0	0	0	0	0	0	0	1	6518	0	0	0	32
slk	0	74	1	1	1	0	8	1	0	5	0	8	1	0	7328	2	5	0
spa	0	6	1	1	4	10	2	6	0	0	1	1	34	1	0	5866	0	0
swe	1	0	31	0	5	2	2	0	0	0	12	1	0	0	3	6	5890	0
ukr	0	0	0	0	0	0	0	0	0	0	0	0	0	21	0	0	0	6911

MUNI FI

	arb	ces	dan	deu	eng	fra	hat	ita	jpn	lin	nld	pol	por	rus	slk	spa	swe	ukr
arb	99,98%	0,00%	0,00%	0,00%	0,00%	0,02%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%
ces	0,00%	98,11%	0,01%	0,10%	0,06%	0,00%	0,03%	0,01%	0,00%	0,03%	0,00%	0,10%	0,06%	0,00%	1,44%	0,00%	0,01%	0,01%
dan	0,00%	0,00%	98,82%	0,08%	0,00%	0,05%	0,10%	0,05%	0,00%	0,00%	0,17%	0,05%	0,02%	0,00%	0,00%	0,00%	0,66%	0,00%
deu	0,00%	0,00%	0,07%	99,48%	0,09%	0,05%	0,05%	0,00%	0,00%	0,00%	0,09%	0,02%	0,00%	0,00%	0,02%	0,07%	0,07%	0,00%
eng	0,00%	0,07%	0,07%	0,00%	99,12%	0,13%	0,02%	0,04%	0,00%	0,02%	0,22%	0,07%	0,04%	0,00%	0,18%	0,02%	0,00%	0,00%
fra	0,00%	0,05%	0,13%	0,00%	0,03%	99,11%	0,02%	0,05%	0,00%	0,08%	0,10%	0,00%	0,06%	0,00%	0,15%	0,23%	0,00%	0,00%
hat	0,00%	0,03%	0,04%	0,00%	0,01%	0,03%	99,43%	0,03%	0,00%	0,16%	0,06%	0,09%	0,01%	0,00%	0,06%	0,06%	0,00%	0,00%
ita	0,00%	0,00%	0,00%	0,07%	0,07%	0,28%	0,03%	98,86%	0,00%	0,05%	0,02%	0,08%	0,12%	0,00%	0,07%	0,33%	0,03%	0,00%
jpn	0,00%	0,00%	0,00%	0,00%	0,06%	0,00%	0,01%	0,00%	99,93%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%
lin	0,00%	0,04%	0,04%	0,01%	0,05%	0,00%	0,09%	0,04%	0,00%	99,49%	0,01%	0,07%	0,02%	0,01%	0,06%	0,04%	0,02%	0,00%
nld	0,00%	0,00%	0,50%	0,13%	0,05%	0,03%	0,07%	0,02%	0,03%	0,05%	98,79%	0,03%	0,02%	0,00%	0,13%	0,05%	0,08%	0,00%
pol	0,00%	0,07%	0,01%	0,03%	0,03%	0,00%	0,04%	0,01%	0,00%	0,06%	0,01%	99,50%	0,00%	0,00%	0,23%	0,00%	0,00%	0,00%
por	0,00%	0,04%	0,00%	0,04%	0,16%	0,02%	0,02%	0,04%	0,00%	0,02%	0,18%	0,05%	98,42%	0,00%	0,11%	0,93%	0,00%	0,00%
rus	0,00%	0,00%	0,00%	0,02%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,02%	99,48%	0,00%	0,00%	0,00%	0,49%
slk	0,00%	1,00%	0,01%	0,01%	0,01%	0,00%	0,11%	0,01%	0,00%	0,07%	0,00%	0,11%	0,01%	0,00%	98,56%	0,03%	0,07%	0,00%
spa	0,00%	0,10%	0,02%	0,02%	0,07%	0,17%	0,03%	0,10%	0,00%	0,00%	0,02%	0,02%	0,57%	0,02%	0,00%	98,87%	0,00%	0,00%
swe	0,02%	0,00%	0,52%	0,00%	0,08%	0,03%	0,03%	0,00%	0,00%	0,00%	0,20%	0,02%	0,00%	0,00%	0,05%	0,10%	98,94%	0,00%
ukr	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,30%	0,00%	0,00%	0,00%	99,70%

MUNI FI

Examples of mistakes

- Czech labeled as Slovak

Často majú viac autonómie ako konvenční členovia tímu, pretože které se rozbíjejí o pláž **často** ich tímy sa môžu stretávať Ostseekreuzfahrten umfassen einen längeren Aufenthalt in Petersburg podľa meniacich sa časových pásiem, čo nemusí byť chápané miestnym manažmentom.

Často majú viac autonómie ako konvenční členovia tímu, pretože které se rozbíjejí o pláž **často** ich tímy sa môžu stretávať Ostseekreuzfahrten umfassen einen längeren Aufenthalt in Petersburg podľa meniacich sa časových pásiem, čo nemusí byť chápané miestnym manažmentom.

Examples of mistakes

- Czech labeled as Slovak

バチカン 市 国 で 働く **na lety z bodu A do** 人々 ozwa viza mosusu ya masuwa mosusu elakisaka の ほとんど が 日常 的 に 使用 する 言語 は イタリア 語 で あり 、宗教 的 な 儀式 で は ラテン 語 が よく 使用 さ れ て い ます 。

バチカン 市 国 で 働く **na lety z bodu A do** 人々 ozwa viza mosusu ya masuwa mosusu elakisaka の ほとんど が 日常 的 に 使用 する 言語 は イタリア 語 で あり 、宗教 的 な 儀式 で は ラテン 語 が よく 使用 さ れ て い ます 。

MUNI FI

Examples of mistakes

- Czech labeled as Slovak

Marocký sultán finland er et fantastisk rejsemål for mesto prestaval a pomenoval ho Dár al-Bajdá a španielski obchodníci, ktorí v meste založili obchodné základne, mu dali meno signálem může být satelitní telefon **vaší** Casablanca .

Marocký sultán finland er et fantastisk rejsemål for mesto prestaval a pomenoval ho Dár al-Bajdá a španielski obchodníci, ktorí v meste založili obchodné základne, mu dali meno signálem může být satelitní telefon **vaší** Casablanca .

Examples of mistakes

- Czech labeled as Slovak

Это предоставляет нам ve **vzdálených polohách bez pokrytí mobilním** большой объем данных и материалов для создания имитационных моделей, помогающих понять процессы, происходящие в 必ず しも ヨット нашем сознании.

Это предоставляет нам ve **vzdálených polohách bez pokrytí mobilním** большой объем данных и материалов для создания имитационных моделей, помогающих понять процессы, происходящие в 必ず しも ヨット нашем сознании.

Examples of mistakes

- Czech labeled as Slovak

غوما هي مدينة سياحية في багатьох випадках навчання جمهورية الكونغو الديمقراطية تقع في أقصى الشرق بالقرب je to v Norsku Švédsku a na من رواندا .

غوما هي مدينة سياحية في багатьох випадках навчання جمهورية الكونغو الديمقراطية تقع في أقصى الشرق بالقرب je to v Norsku Švédsku a na من رواندا .

Examples of mistakes

- Slovak labeled as Czech

I i-länder hör man sällan liknande klagomål om vattenkvalitet eller **nezachytí prípadne v** أو تعرضوا إنه ليس منه فهو لا **závislosti od vašich zručností** broar som rasar.

I i-länder hör man sällan liknande klagomål om vattenkvalitet eller **nezachytí prípadne v** أو تعرضوا إنه ليس منه فهو لا **závislosti od vašich zručností** broar som rasar.

MUNI FI

Examples of mistakes

- Slovak labeled as Czech

Na a rôzne pyramídy sú **osvetlené ostrovy** severu a few deeper sections je oblast ohraničená Sahelem, na jihu a západě pak Atlantským oceánem.

Na a rôzne pyramídy sú **osvetlené ostrovy** severu a few deeper sections je oblast ohraničená Sahelem, na jihu a západě pak Atlantským oceánem.

MUNI FI

Examples of mistakes

- Slovak labeled as Czech

Starożytni Egipcjanie z okresu Nowego enough for any yacht smaller boats or **cestách mohla vykládka** Państwa patrzyli z podziwem na budowle swoich przodków, które liczyły sobie wtedy ponad tysiąc lat.

Starożytni Egipcjanie z okresu Nowego enough for any yacht smaller boats or **cestách mohla vykládka** Państwa patrzyli z podziwem na budowle swoich przodków, które liczyły sobie wtedy ponad tysiąc lat.

Examples of mistakes

- English labeled as French

The normal (local) price is ~500 **Congolese Francs** données à distance et de téléphonie votre for が あっ た the short ride.

The normal (local) price is ~500 **Congolese Francs** données à distance et de téléphonie votre for が あっ た the short ride.

Examples of mistakes

- French labeled as English

Lorsque toutes les ressources disponibles sont utilisées efficacement dans les services fonctionnels d'une it is not **organisation**, la créativité et l'ingéniosité peuvent s'exprimer pleinement.

Lorsque toutes les ressources disponibles sont utilisées efficacement dans les services fonctionnels d'une it is not **organisation**, la créativité et l'ingéniosité peuvent s'exprimer pleinement.

Outline

- Description of the task and the model (dataset, training, evaluation)
- Practical applications
- Optimizing input processing

Practical Applications

- Data labeling
  - Language identification (rapcor)
- Data filtering
  - Filtering data for machine translation

Problems with filtering monolingual data

- Filters short sentences
- Filters sentences written in non-literal language

Examples of problematic filtered sentences

- All tokens of a Czech sentence labeled as Slovak:
  - To je ono.
  - Jamesi, já nestojím o hrdinu.
  - To je v poho, no tak.
  - Drž kurva hubu!
  - Já ti ukážu, co je to pravý muž.
  - Dobrá práce, Gusy.
  - A taky mi pomohl zapomenout na Paka.

Examples of problematic filtered sentences

- All tokens of a Czech sentence labeled as a different language:
  - No a? - Portuguese
  - Zrzek. - Polish
  - Hele. - Danish
  - Omdlela. - Swedish

Outline

- Description of the task and the model (dataset, training, evaluation)
- Practical applications
- Optimizing input processing

Optimizing input processing

- speed improvement options:
  - Loading the model just once
  - Processing multiple inputs simultaneously
  - Using GPUs

```
def langtok_model(input: str): -> List[str]
    model = AutoModelForTokenClassification.from_pretrained(langtok)
    tokenizer = AutoTokenizer.from_pretrained(langtok)
    result = model(input)
    return result

def main():
    sentences =
    for sentence in sentences:
        result = langtok_model(sentence)
```


MUNI FI

Optimizing input processing

- Loading the model just once

```
def langtok_model(input: str, model, tokenizer): -> List[str]
    [redacted]
    [redacted]
    result = model(input)
    [redacted]
    [redacted]
    return result
```

```
def main():
    model = AutoModelForTokenClassification.from_pretrained(langtok)
    tokenizer = AutoTokenizer.from_pretrained(langtok)
    [redacted]
    sentences = [redacted]
    for sentence in sentences:
        result = langtok_model(sentence, model, tokenizer)
    [redacted]
    [redacted]
    [redacted]
```


MUNI FI

Optimizing input processing

- Processing multiple inputs simultaneously

```
def langtok_model(inputs: List[str]): -> List[List[str]]
    model = AutoModelForTokenClassification.from_pretrained(langtok)
    tokenizer = AutoTokenizer.from_pretrained(langtok)
    [redacted]
    results = model(inputs)
    [redacted]
    [redacted]
    return results
```

```
def main():
    [redacted]
    sentences = [redacted]
    results = langtok_model(sentences)
    [redacted]
    [redacted]
    [redacted]
```


MUNI FI

Optimizing input processing

- Processing multiple inputs simultaneously

```
def langtok_model(inputs: List[str], model, tokenizer): -> List[List[str]]
    [redacted]
    [redacted]
    results = model(inputs)
    [redacted]
    [redacted]
    return results
```

```
def main():
    model = AutoModelForTokenClassification.from_pretrained(langtok)
    tokenizer = AutoTokenizer.from_pretrained(langtok)
    [redacted]
    sentences = [redacted]
    results = langtok_model(sentences, model, tokenizer)
    [redacted]
    [redacted]
    [redacted]
```


MUNI FI

Optimizing input processing

- Using GPUs

```
def langtok_model(inputs: List[str], model, tokenizer): -> List[List[str]]
  tokenized_inputs = tokenizer(inputs)
  if  torch.cuda.is_available():
    model.to(torch.device("cuda"))
    tokenized_inputs = tokenized_inputs.to(torch.device("cuda"))
  results = model(inputs)
  return results
```

```
def main():
  model = AutoModelForTokenClassification.from_pretrained(langtok)
  tokenizer = AutoTokenizer.from_pretrained(langtok)
  sentences =
  results = langtok_model(sentences, model, tokenizer)
```


MUNI FI

Optimizing input processing

using GPU	loading model only once	simultaneous processing	duration of processing [s]	
			short file	long file
			14,421	107,867
			14,493	128,564
			1,886	5,892
			3,905	5,018
			1,582	3,67
			3,241	4,591
			1,616	3,848
			3,152	4,566

machine	apollo
GPU type	Tesla T4
short file length [sentences]	20
long file length [sentences]	200

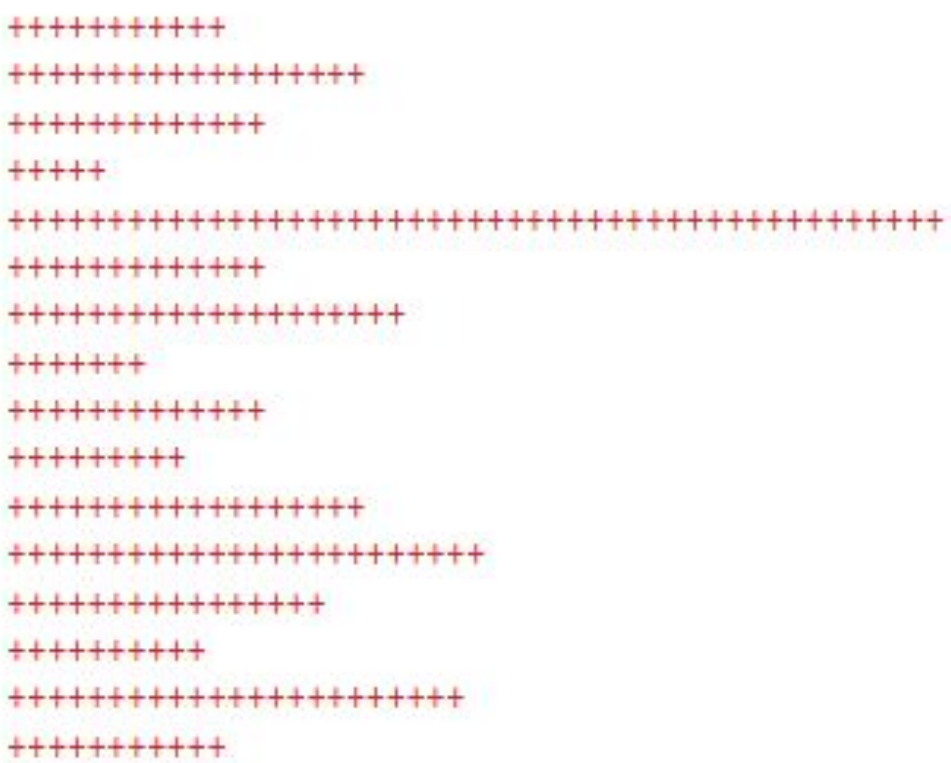
MUNI FI

Batch Size Value For Processing Inputs

- Processing multiple inputs at once increases the speed
- The inputs have to be padded to the same length
- Multiplying matrices that are too large is computationally expensive

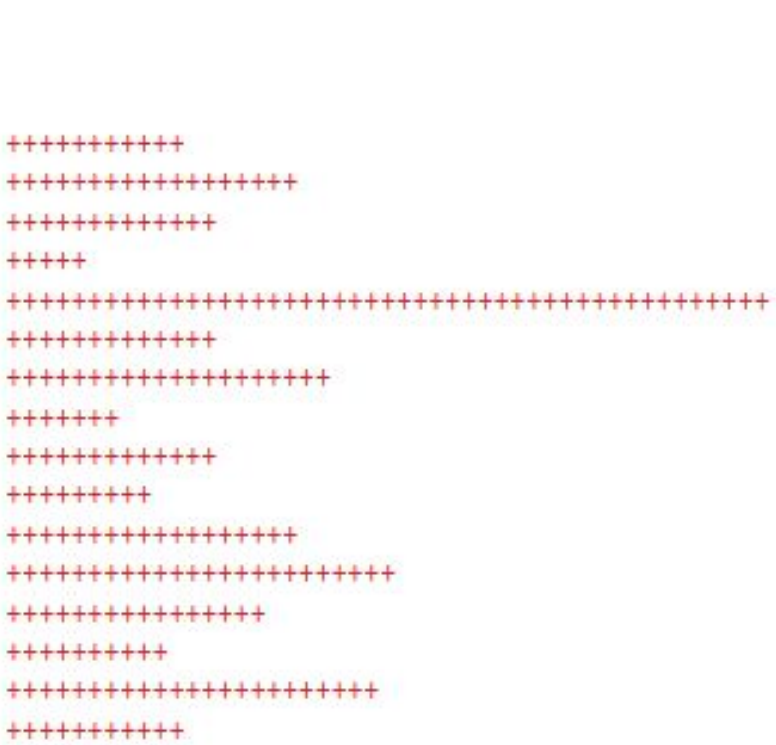
- Balance the benefit of speeding up computation by processing multiple inputs simultaneously with the cost of working with large matrices

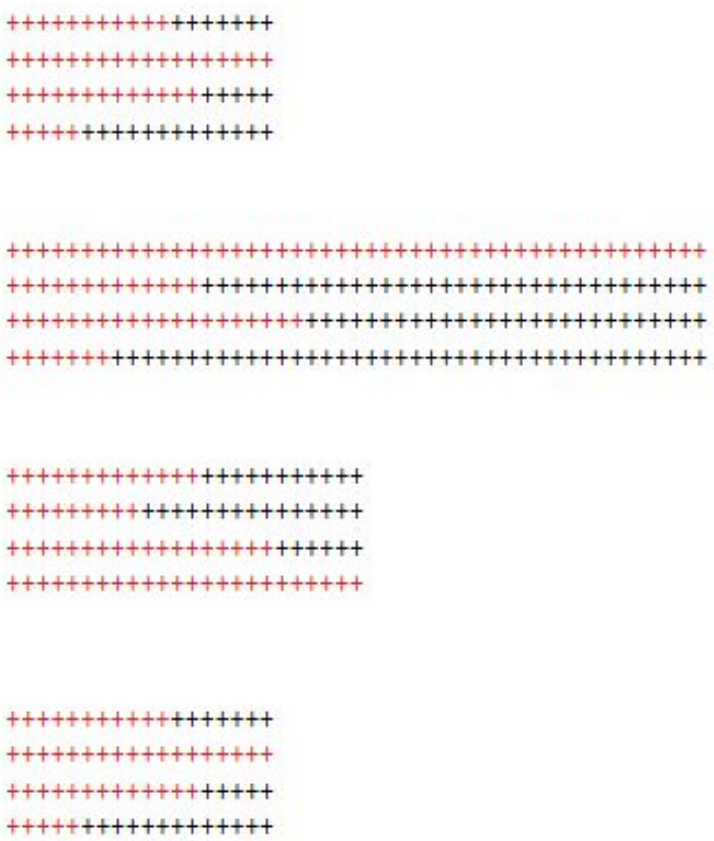
Batch Size Value For Processing Inputs



MUNI FI

Batch Size Value For Processing Inputs



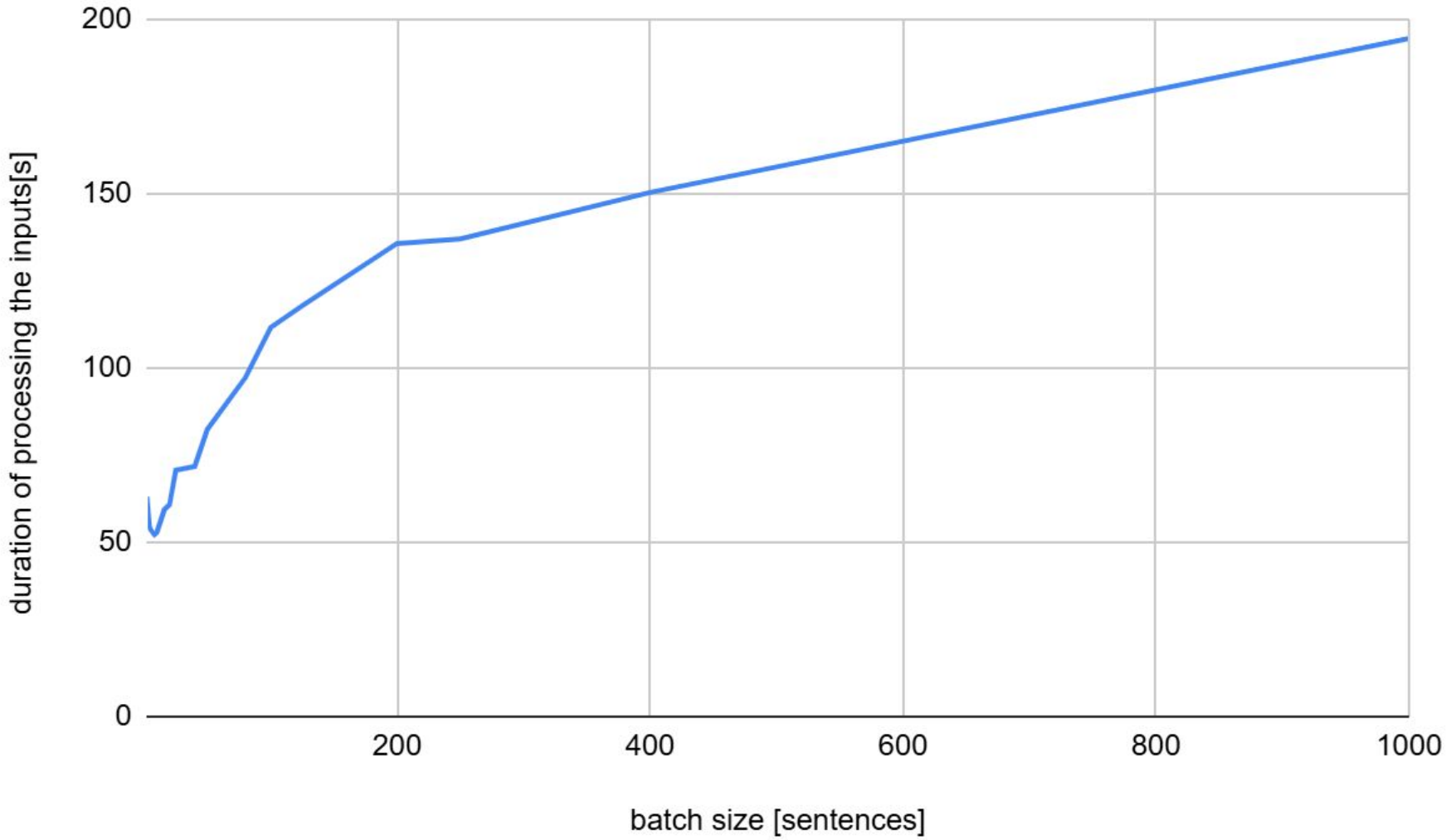


MUNI FI

batch size [sentences]	duration of loading the model [s]	duration of processing the inputs[s]	overall duration [s]	min batch length	avg batch length	avg memory [GB]	avg %CPU
2	3,512	63,092	66,605	4	29,644	1,071	1602,000
4	2,348	54,779	57,127	5	38,246	1,150	1607,417
5	2,231	53,678	55,910	5	41,823	1,150	1607,111
8	2,218	52,240	54,458	5	49,368	1,167	1608,667
10	1,912	52,816	54,729	6	54,515	1,221	1607,526
16	2,853	59,403	62,256	7	63,896	1,265	1605,706
20	2,537	60,770	63,307	12	71,890	1,294	1604,944
25	4,412	70,700	75,112	17	76,863	1,300	1603,294
40	2,418	71,787	74,205	20	91,520	1,538	1602,615
50	2,386	82,390	84,776	29	102,100	1,463	1592,375
80	2,588	97,209	99,797	58	121,080	1,464	1583,880
100	2,474	111,648	114,123	58	130,650	1,327	1563,400
125	2,644	117,890	120,534	63	137,250	1,341	1545,200
200	2,629	135,769	138,399	119	156,900	2,511	1501,474
250	2,683	137,066	139,749	63	156,875	1,779	1460,632
400	2,702	150,435	152,810	128	172,200	2,326	1438,421
500	2,621	157,735	160,357	119	180,500	2,800	1443,053
1000	2,694	194,586	197,280	204	216,500	5,558	1384,526

machine	epimetheus4
source length [sentences]	2000

MUNI FI

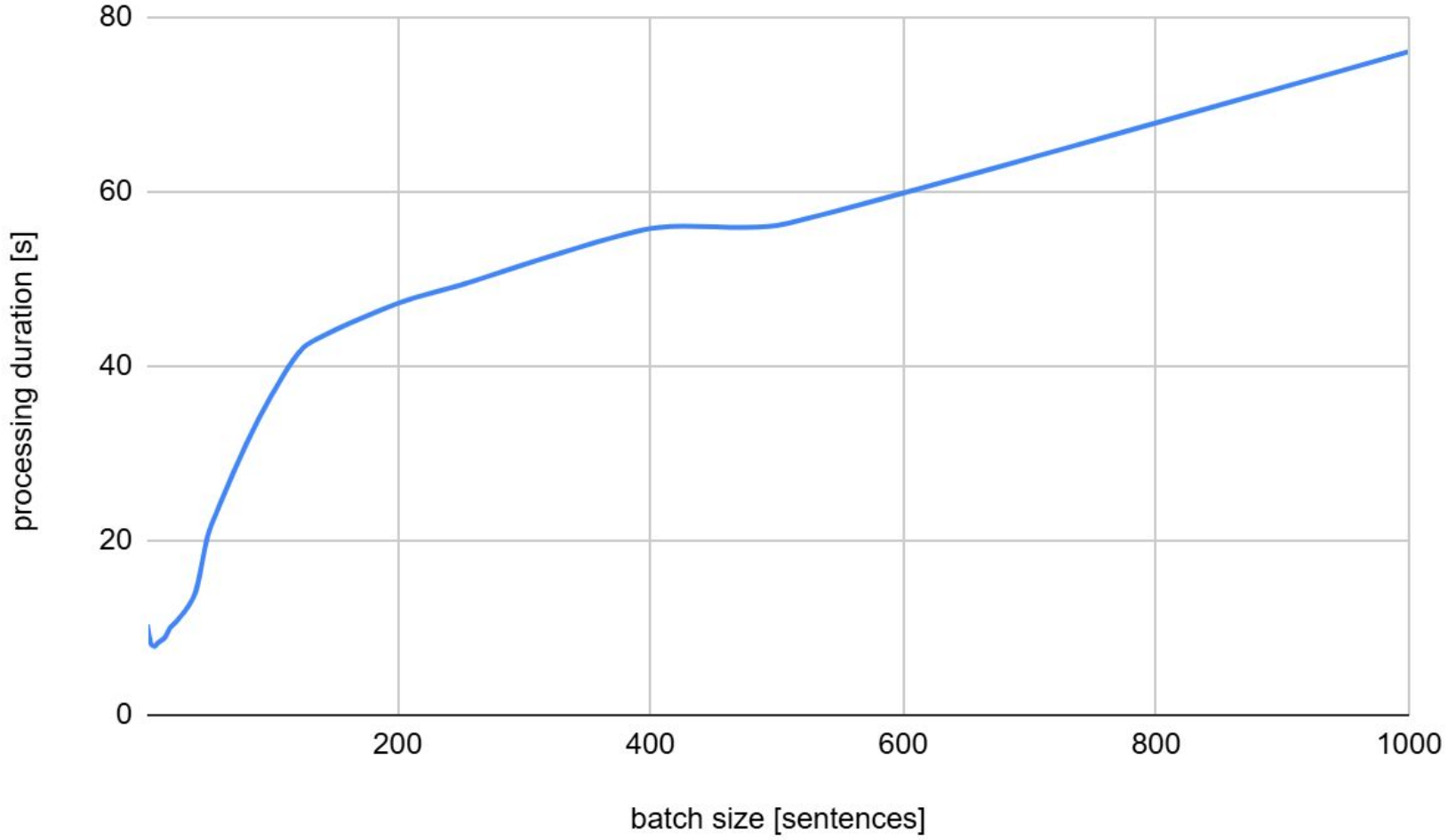

200
150
100
50
0
duration of processing the inputs[s]
200
400
600
800
1000
batch size [sentences]

MUNI FI

batch size [sentences]	duration of loading the model [s]	duration of moving the model to GPU [s]	duration of processing the inputs [s]	overall duration [s]	min batch length	avg batch length	avg %GPU	max %GPU	CPU max memory	max %CPU	GPU memory [MB]
2	1,851	1,019	10,373	13,244	4	29,644	52,2	72	810440	156,4	1313
4	1,327	0,633	8,644	10,605	5	38,246	55,5	77	801916	211,6	1345
5	1,383	0,643	8,112	10,138	5	41,8225	57,2	73	802912	209,6	1361
8	1,545	0,632	7,895	10,073	5	49,368	63,7	77	803616	203,5	1399
10	1,470	0,651	8,231	10,354	6	54,515	69,9	86	803064	189,4	1421
16	1,449	0,652	8,920	11,022	7	63,896	76,9	87	803852	172,6	1525
20	1,414	0,631	10,012	12,059	12	71,89	77,1	86	807108	210,6	1571
25	1,700	0,668	10,743	13,111	17	76,8625	79,4	92	802548	152,6	1627
40	1,268	0,635	14,040	15,944	20	91,52	78,1	100	806960	137,0	1853
50	1,695	0,667	20,661	23,024	29	102,1	74,4	100	824864	130,7	2011
80	1,574	0,690	31,012	33,277	58	121,08	77,6	100	802544	214,9	2429
100	1,518	0,685	36,633	38,836	58	130,65	74,5	100	802972	116,2	2507
125	1,553	0,703	42,096	44,354	63	137,25	81,6	100	804344	116,4	2801
200	1,513	0,697	47,218	49,428	119	156,9	86,4	100	808352	110,2	3719
250	1,572	0,713	49,327	51,613	63	156,875	85,8	100	808564	112,2	4313
400	2,068	0,702	55,811	58,581	128	172,2	79,8	100	815188	113,6	6133
500	1,960	0,688	56,149	58,797	119	180,5	82,4	100	815660	112,2	7333
1000	1,896	0,704	76,107	78,708	204	216,5	84,7	100	836120	118,2	13373

machine	epimetheus2
source length [sentences]	2000
GPU type	Tesla T4

MUNI FI

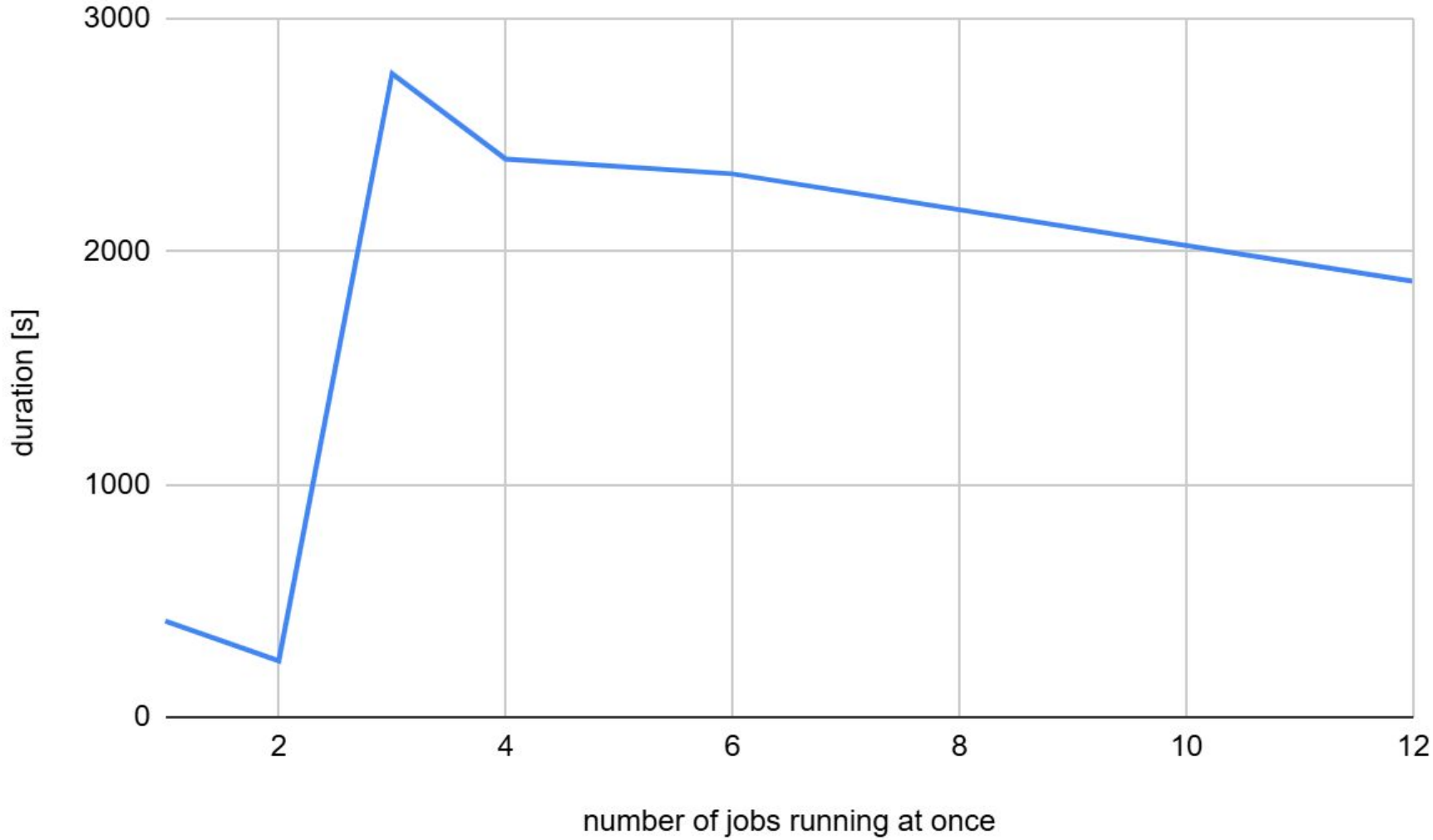

80
60
40
20
0
processing duration [s]
200
400
600
800
1000
batch size [sentences]

Running more processes simultaneously

number of jobs running at once	duration [s]	average duration of processing one file [s]	CPU memory values / process	%CPU values / process
1	413	12,907	1,0g - 1,3g	1300 - 1620
2	242	16,934	1,0g - 1,1g	1560 -1606
3	2764	341,365	970m - 1,1g	950 - 1160
4	2398	395,119	914m - 1,1g	750 - 840
6	2334	578,112	880m - 1,0g	507 - 572
12	1872	927,829	959m - 1,0g	256 - 277

machine	epimetheus1
number of files to process	24
file length [sentences]	400

MUNI FI

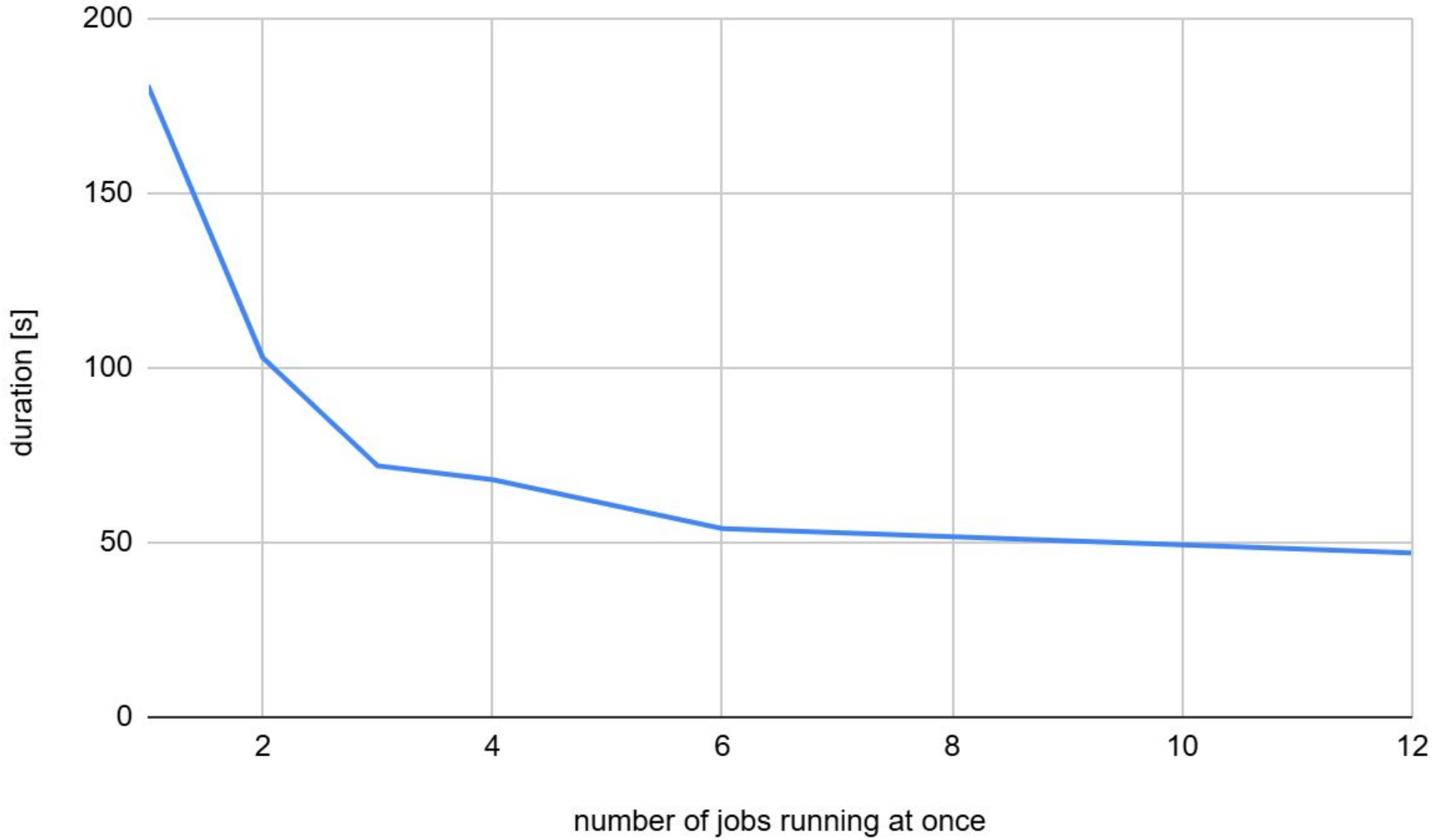

3000
2000
1000
0
duration [s]
2
4
6
8
10
12
number of jobs running at once

Running more processes simultaneously

number of processes running simumltaneously	duration [s]	average duration of processing one file [s]	avg GPU temperature [°C]	avg %GPU	GPU memory [MB]	max CPU memory / process	max %CPU values / process
1	181	4,325	56,191	34,2	1291,8	1g	198,0
2	103	5,260	54,466	58,4	2232,5	800488m	196,7
3	72	5,478	58,983	57,8	2455,8	800780m	209,6
4	68	7,498	59,526	79,7	3896,4	1,1g	210,9
6	54	9,091	59,464	75,2	4614,5	802216	194,0
12	47	17,497	53,721	92,0	10053,9	801100	113,2

machine	epimetheus1
number of files to process	24
file length [sentences]	400
GPU type	Tesla T4

MUNI FI


200
150
100
50
0
duration [s]
2
4
6
8
10
12
number of jobs running at once

MUNI
FI

Room for discussion

Links and Resources

- Token Classification on Hugging Face (tutorial)
- FLORES-200 (dataset)
- LANGTOK Hugging Face
- Datasets on Hugging Face: train, validate, test
- BERT multilingual base model (cased)
- Demo Tokenized Hugging Face, Demo Hugging Face

Thank You For Your Attention

MUNI
FI