kaggle

2nd Place Solution
Learning Equality Challenge

Konrad Habel

Agenda

Agenda

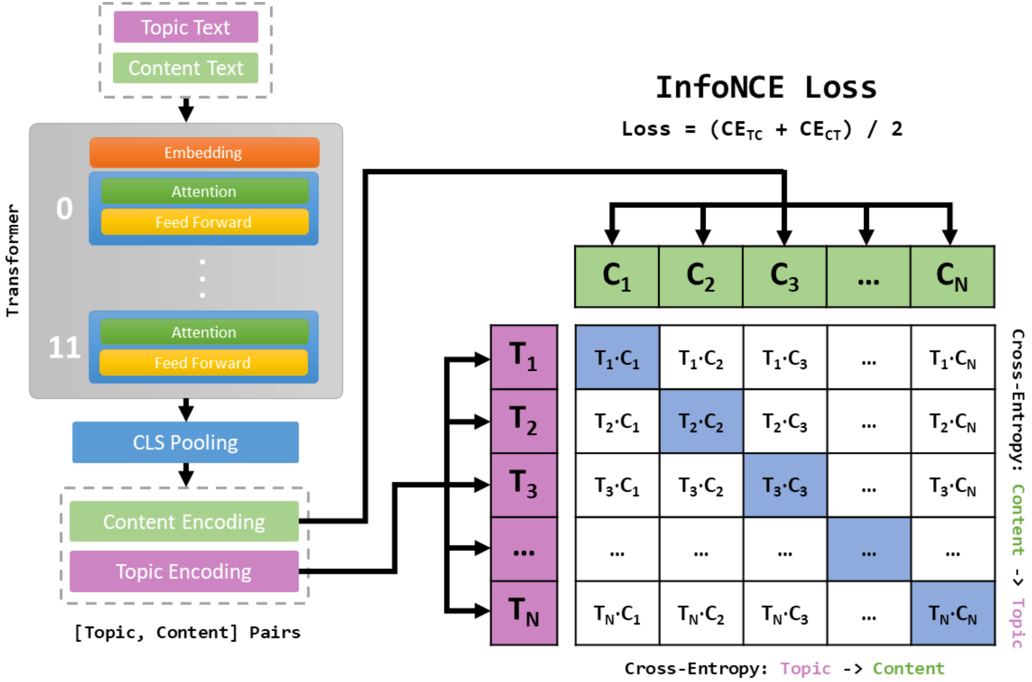
1. Background
2. Approach
3. Language Switching
4. Knowledge Distillation
5. Quantization
6. Dynamic Threshold
7. Results
8. Important and Interesting Findings
9. Question and Answer

Background

- Ph.D. student at the University of the Bundeswehr Munich in Germany
- German diploma in Automotive Engineering
- Bachelor of Science (B.Sc.) in Business & Information Systems Engineering
- Master of Science (M.Sc.) in Computer Science

Approach

- Solution is a **single stage approach** based on cosine similarity for retrieval using the **InfoNCE** loss as training objective.



Input Data: No use of special token for separation instead, the **#** is used as separator.

Topic: Title **#** Topic-Tree **#** Description

The topic tree is reverse ordered and the same separator **#** is used:

-> Title **#** Parent **#** Grandparent **#** ... **#** Description

Content: Title **#** Description **#** Text (cut to 32 based on white space splitting)

If for example the Description is empty the model will see as input:

-> Title **#** **#** Text

For both Topic and Content a maximum sequence length of 96 tokens is used.

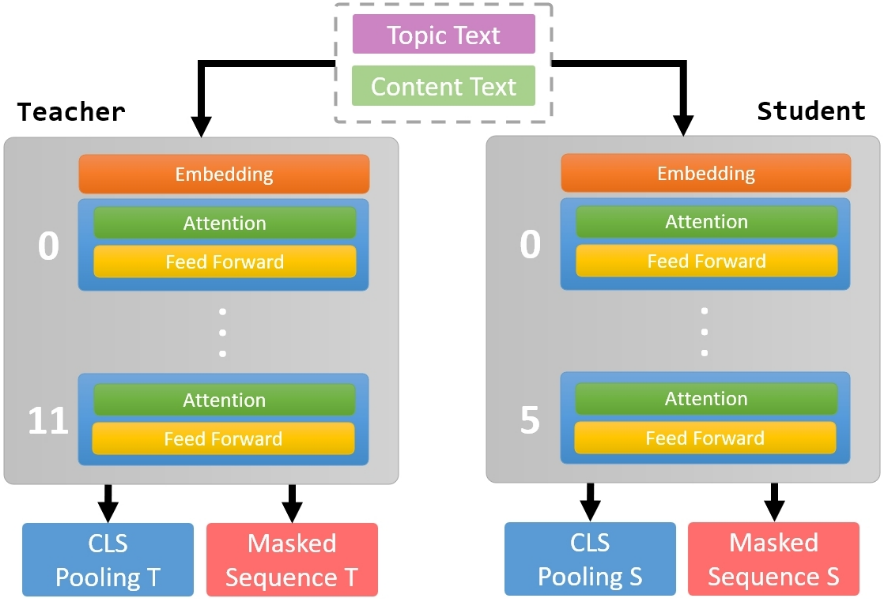
Language Switching

- Translation of the most common languages **en, es, pt, fr** into each other for additional training data.

Knowledge Distillation

Knowledge Distillation

- Distillation from 12 -> 6 Layers used for the Efficiency Prize




```
initial layer/weight transfer
Teacher -> Student
(1,3,5,7,9,11) -> (0,1,2,3,4,5)
```

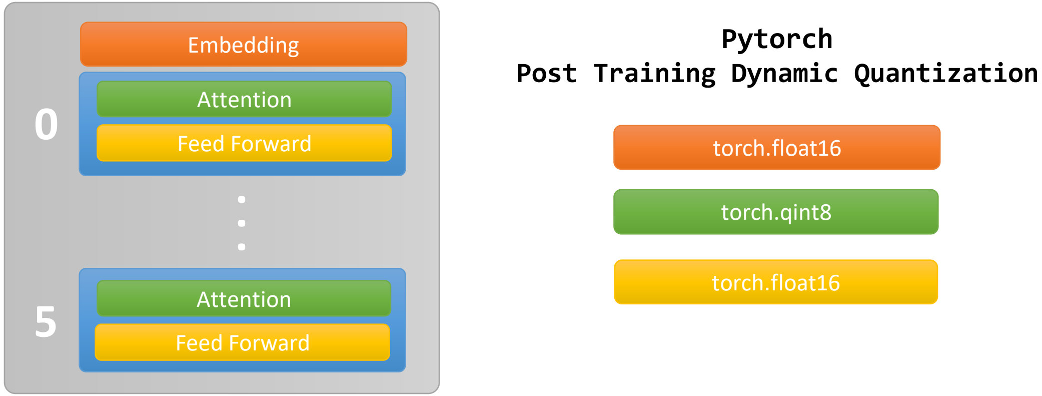
```
MSE_CLS = MSE_LOSS( CLS Pooling T , CLS Pooling S )

MSE_SEQ = MSE_LOSS( Masked Sequence T , Masked Sequence S )

Loss = (MSE_CLS + MSE_SEQ) / 2
```

Quantization

- Using Pytorch “Post Training Dynamic Quantization” for the Efficiency Prize



Dynamic Threshold

Dynamic Threshold

Cosine Similiarity Matrix [per Language]

	C_1	C_2	C_3	…	C_n
T_1					
T_2					
T_3					
…					
T_i					

Calculation for each Topic [row]:

```
margin = 0.16
```

```
content_ids_l = content_ids[language]
```

```
sim = sim_matrix[i]

dyn_th = sim.max() – margin * sim.max()

p_select = sim >= dyn_th

c_choice = content_ids_l[p_select]
```

Topic with **low** max similarity scores

```
dyn_th = 0.4 – 0.16 * 0.4 = 0.336

sim_score_window = [0.336 : 0.4]
```

-> less likely to pick content next to the **content** with highest similiarity for that **topic**

Topic with **high** max similarity scores

```
dyn_th = 0.8 – 0.16 * 0.8 = 0.672

sim_score_window = [0.672 : 0.8]
```

-> more likely to pick other content next to the **content** with highest similarity for that **topic**

- for each Topic the size of the window for potential Content depends on the best retrieved candidate
- no good static threshold for all data must be found
- only the margin must be selected but is not so sensitive than a static threshold

-> at least one Content is always retrieved no matter how low the max. similarity score is

Results

- Influence of the distillation and quantization:

Table 1: Results for sentence-transformers/LaBSE trained on fold 0 with margin=0.16.

Model	Fold 0	Public	Privat	Run-Time (test-set)
12 layer	0.6660	0.6637	0.7026	P100 4 min
6 layer (distilled)	0.6631	0.6523	0.6907	P100 3 min
6 layer (distilled + quantized)	0.6609	0.6526	0.6895	CPU 13 min

models are trained on a single RTX 3090 for 40 epochs

- Combining models into an ensemble:

Table 2: Results for models trained on fold 0 with margin=0.16.

Model	Fold 0	Public	Privat
sentence-transformers/LaBSE	0.6660	0.6637	0.7026
sentence-transformers/paraphrase-multilingual-mpnet-base-v2	0.6615	0.6618	0.7092
ensemble (50%/50%)	0.6849	0.6808	0.7238

models are trained on a single RTX 3090 for 40 epochs

Fold 0

- Results per language:

```
------------------------------[Ensemble]------------------------------
en   Score: 0.68640 - Precision: 0.65308 - Recall: 0.742 - Selected:   5 - (2806x65939)
es   Score: 0.76713 - Precision: 0.71482 - Recall: 0.835 - Selected:   4 - (1177x30844)
pt   Score: 0.79943 - Precision: 0.74148 - Recall: 0.860 - Selected:   6 - (343x10435)
ar   Score: 0.54267 - Precision: 0.56559 - Recall: 0.662 - Selected:   4 - (318x7418)
fr   Score: 0.61698 - Precision: 0.64399 - Recall: 0.662 - Selected:   7 - (304x10682)
bg   Score: 0.70410 - Precision: 0.67926 - Recall: 0.756 - Selected:   7 - (242x6050)
bn   Score: 0.17561 - Precision: 0.11966 - Recall: 0.230 - Selected:   7 - (237x2513)
sw   Score: 0.71674 - Precision: 0.65091 - Recall: 0.789 - Selected:   5 - (209x1447)
gu   Score: 0.77115 - Precision: 0.68613 - Recall: 0.839 - Selected:   5 - (181x3677)
hi   Score: 0.71008 - Precision: 0.68468 - Recall: 0.774 - Selected:   7 - (138x4042)
it   Score: 0.87877 - Precision: 0.88017 - Recall: 0.904 - Selected:   4 - (73x1300)
zh   Score: 0.66702 - Precision: 0.59670 - Recall: 0.758 - Selected:   9 - (68x3849)
mr   Score: 0.72963 - Precision: 0.69001 - Recall: 0.798 - Selected:  11 - (24x999)
fil  Score: 0.80584 - Precision: 0.69457 - Recall: 0.882 - Selected:   7 - (23x516)
as   Score: 0.45620 - Precision: 0.31313 - Recall: 0.662 - Selected:   7 - (13x641)
my   Score: 0.76245 - Precision: 0.82500 - Recall: 0.829 - Selected:   3 - (12x206)
km   Score: 0.95030 - Precision: 0.94697 - Recall: 0.958 - Selected:   4 - (11x505)
kn   Score: 0.63193 - Precision: 0.52910 - Recall: 0.744 - Selected:  10 - (9x501)
te   Score: 0.77938 - Precision: 0.54730 - Recall: 0.946 - Selected:  16 - (7x285)
or   Score: 0.72903 - Precision: 0.71429 - Recall: 0.733 - Selected:   8 - (5x326)
ta   Score: 0.68842 - Precision: 0.45114 - Recall: 1.000 - Selected:   9 - (5x216)
ur   Score: 0.35907 - Precision: 0.22956 - Recall: 0.448 - Selected:   9 - (5x245)
pnb  Score: 0.89423 - Precision: 0.82500 - Recall: 0.938 - Selected:   9 - (4x184)
ru   Score: 0.70697 - Precision: 0.67778 - Recall: 0.767 - Selected:  11 - (3x188)
pl   Score: 0.66623 - Precision: 1.00000 - Recall: 0.632 - Selected:  28 - (3x319)
swa  Score: 0.16744 - Precision: 0.11597 - Recall: 0.320 - Selected:  40 - (3x495)
tr   Score: 0.71442 - Precision: 0.87698 - Recall: 0.696 - Selected:  23 - (3x225)
--------------------------------------------------------------------------------
Eval Score: 0.68489 - Precision: 0.64888 - Recall: 0.748
--------------------------------------------------------------------------------
```

Trained on all data

GPU - Ensemble:

```
Ensemble:
Num. Models:      5

Score:
Privat:           0.75479
Public:           0.70977

Inference:
Kernel:           P100 - GPU
Runtime:          9 min
Max. Seq. Length: 96

Trained:
Data:             all data (no folds)
Epochs:           40
Batch Size:       768 – pairs of [Topic, Content]
Seq. Length:      96
LR-Schedule:      polynomial decay with warmup (2 Epoch)
Max. LR:          0.0003
LR-End:           0.0001
GPU:              4xV100 (32GB)

Models:
- 'sentence-transformers/LaBSE'
- 'facebook/mcontriever-msmarco'
- 'sentence-transformers/stsb-xlm-r-multilingual'
- 'sentence-transformers/paraphrase-multilingual-mpnet-base-v2'
- 'sentence-transformers/xlm-r-100langs-bert-base-nli-mean-tokens'
```

CPU - Ensemble:

```
Ensemble:
Num. Models:      2

Score:
Privat:           0.73118
Public:           0.68959

Inference:
Kernel:           CPU
Runtime:          23 min
Max. Seq. Length: 96

Destillation:
Teacher:          pre-trained Model of GPU Submission
Layers keep:      6
Quantization:     post training dynamic
Data:             all data (no folds)
Epochs:           40
Batch Size:       1024 – pairs of [Topic, Content]
Seq. Length:      96
LR-Schedule:      polynomial decay with warmup (2 Epoch)
Max. LR:          0.0003
LR-End:           0.0001
GPU:              4xV100 (32GB)

Models:
- 'sentence-transformers/LaBSE’
- 'sentence-transformers/paraphrase-multilingual-mpnet-base-v2'
```

Important and Interesting Findings

- Models can be trained on consumer-grade hardware like a single RTX 3090 when using gradient-checkpointing.
- Contrastive training with InfoNCE works also for a n x m matching problem when carefully selecting samples for every batch -> sampling strategy plays an important role for good results.
- Influence of the topic tree, maybe training models with different usage of the topic tree information could be useful.

Question and Answer

kaggle