Punctuation Restoration

Presented by:

- Ayoub HAMMAL
- Quentin LE TELLIER
- Junior Cédric TONGA

Introduction

- Automatic Speech Recognition output unsegmented transcripts
  - **Affects readability** as much as high word error rate

- Neural machine translation and sentiment analysis benefit from having clausal boundaries

Data

🤗 Hugging Face BookCorpus

Used to train:

Google's BERT model and its variants: ALBERT, RoBERTa, GPT-N

# of books	# of sentences	# of words	# of unique words	mean # of words/sentence
11_038	74_004_228	984_846_357	1_316_420	13

Data

🤗 Hugging Face BookCorpus

Sets	Train	Validation	Test
# of documents	10_000	1_000	1_000

X

1024 sentences per document

Preprocessing

I am John, from Texas.

↓ Word slicing

["i","am","john",",","from","texas","."]

↓ Punctuation grouping

x	i	am	john	from	texas
y	[EMPTY]	[EMPTY]	[COMMA]	[EMPTY]	[PERIOD]

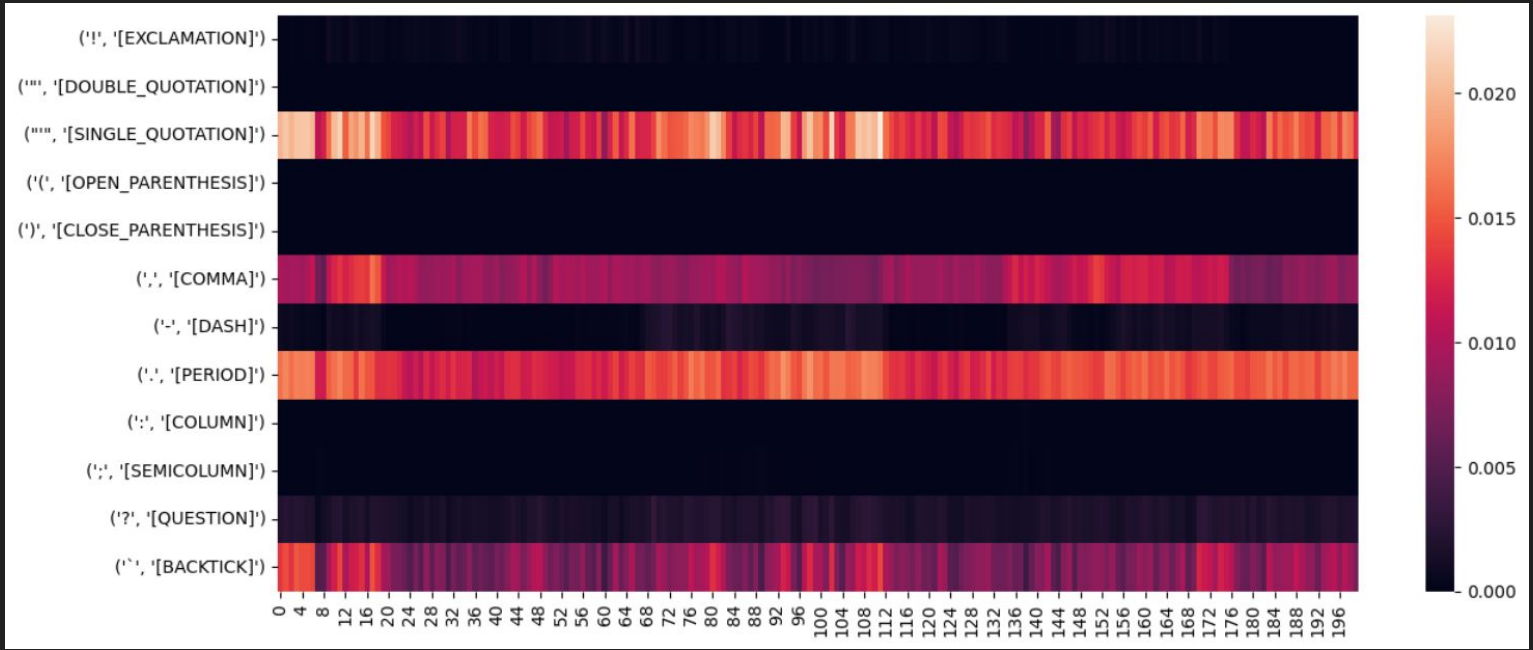
Visualization

Labels distribution
(over the first 1000 documents for each split)

	Train	Validation	Test
[EMPTY]	*9_571_319*	*10_376_878*	*10_476_603*
[PERIOD]	925_569	937_997	936_942
[COMMA]	614_374	648_122	693_574
[QUESTION]	103_364	94_730	97_051

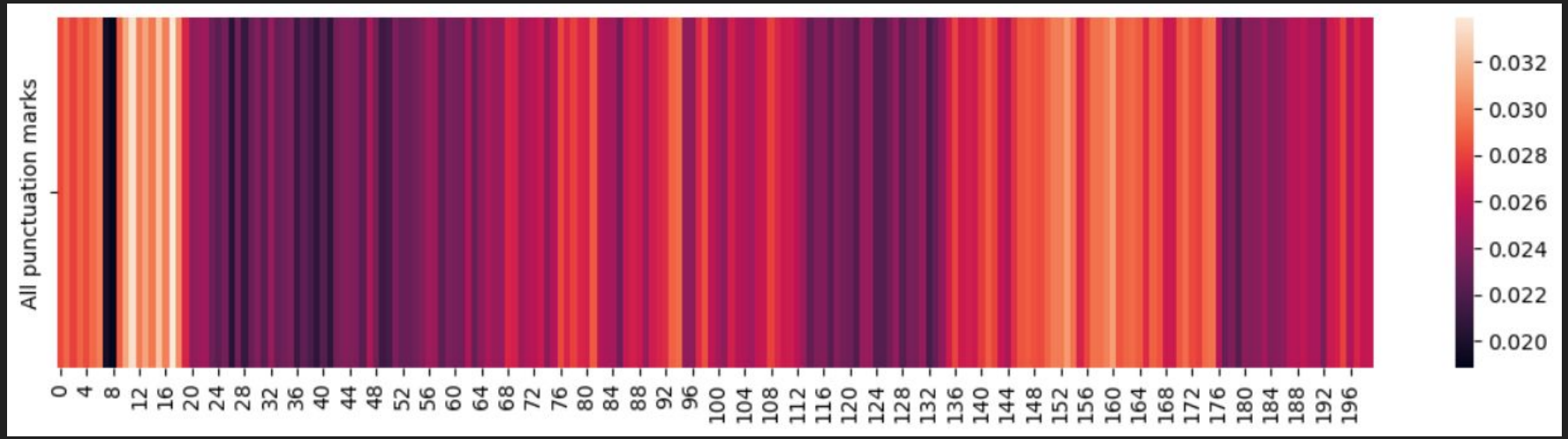
Visualization

Punctuation marks frequencies



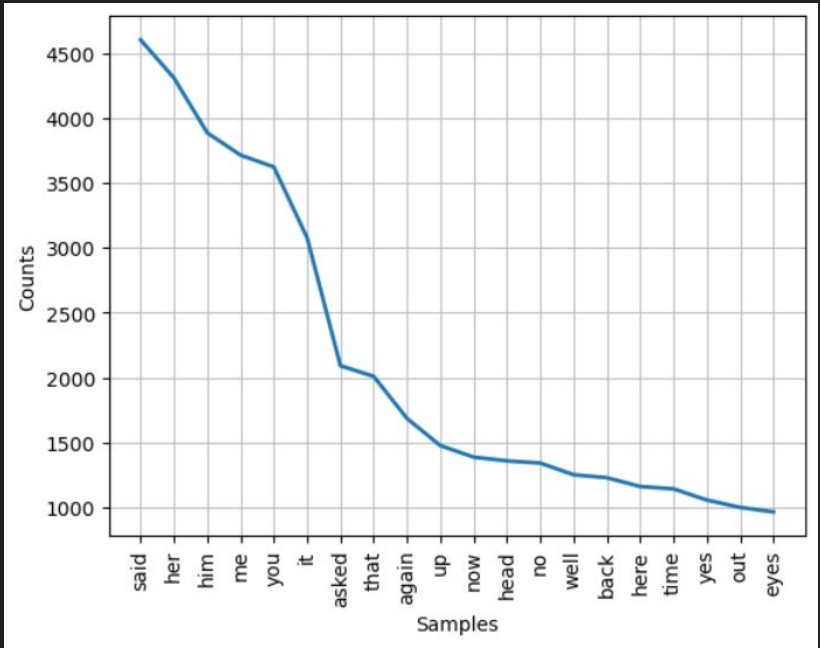
Visualization

Punctuation marks frequencies – All mixed

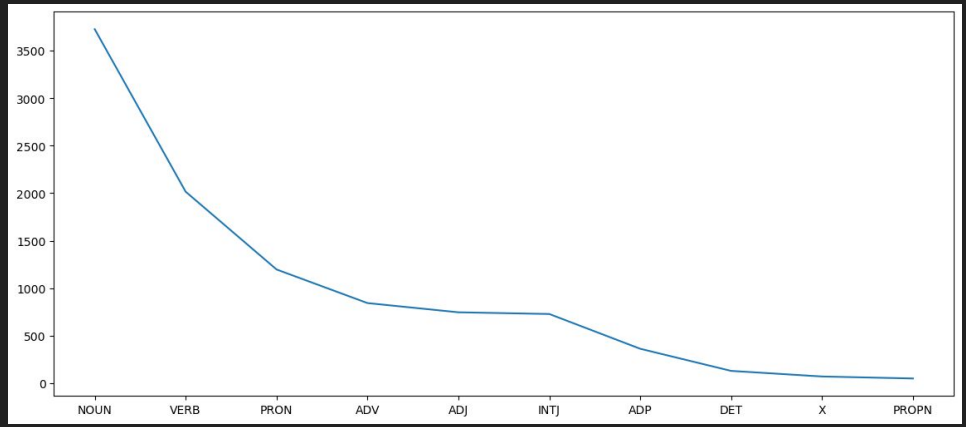


Visualization

Words preceding punctuation marks



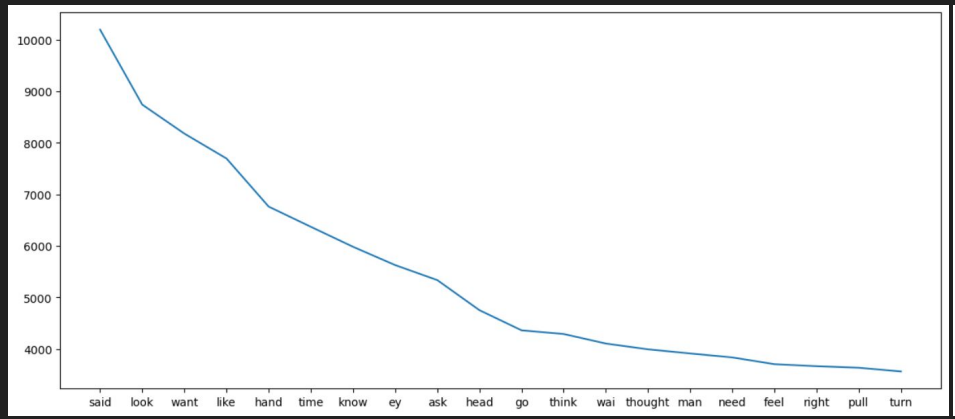
Top most frequent words (Batch of 200_000 sentences)

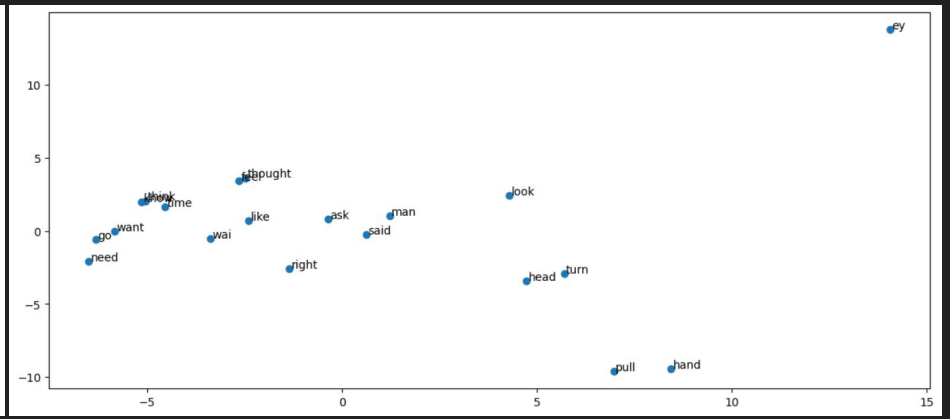


Grammatical class (Batch of 10_000 words)

Visualization

Word embedding visualization



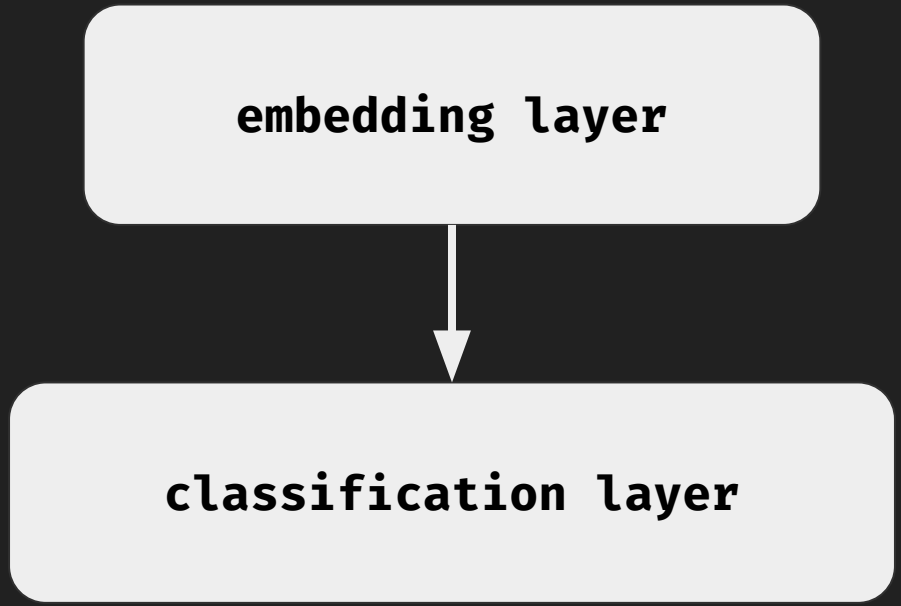


Top 20 most frequent words by occurrence

→

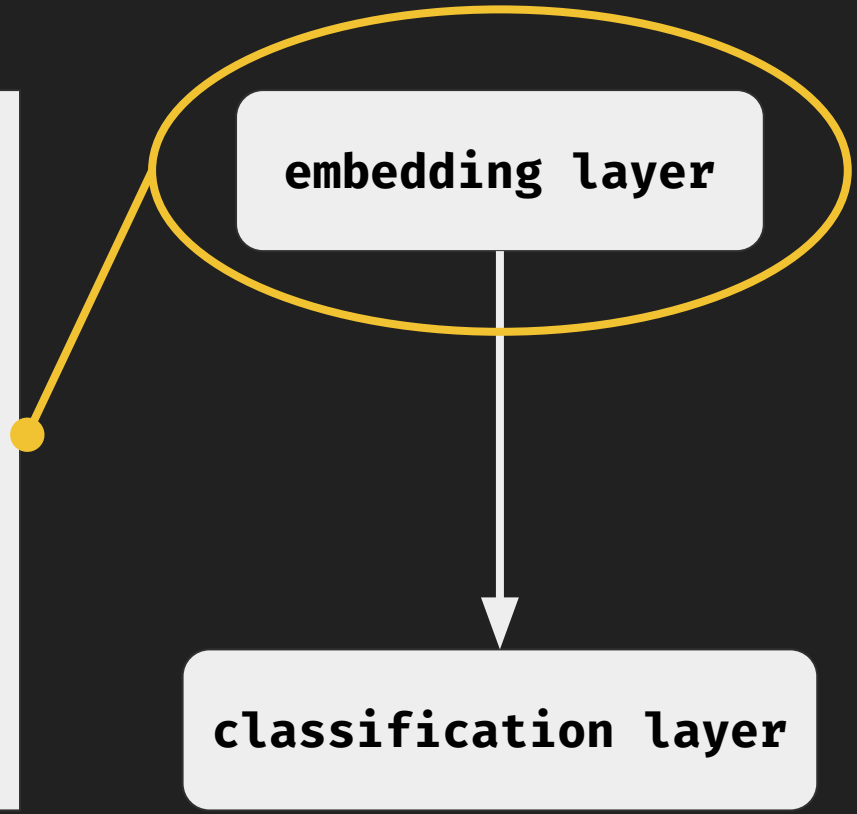
PCA of their word embedding

Training

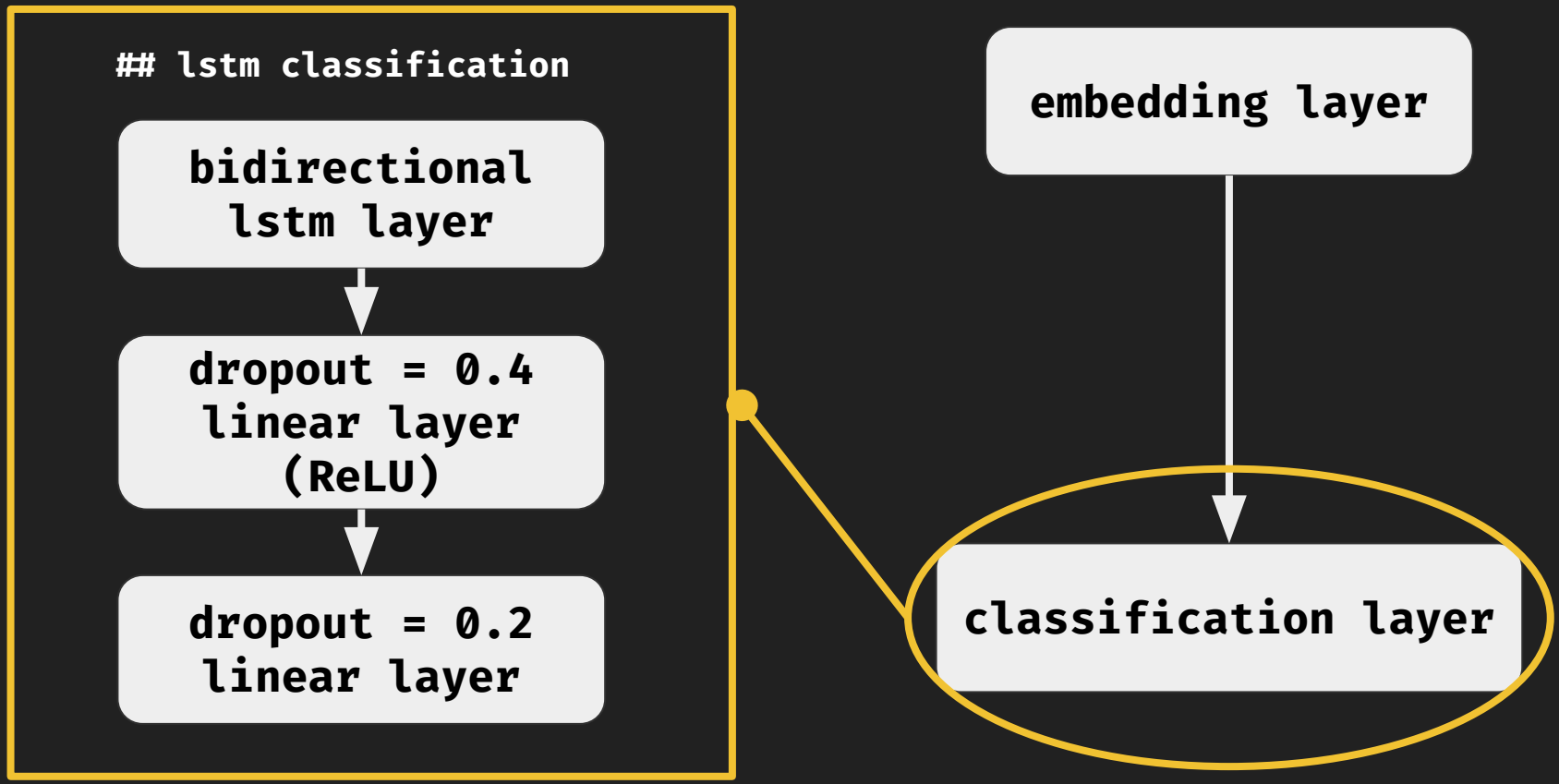


Training

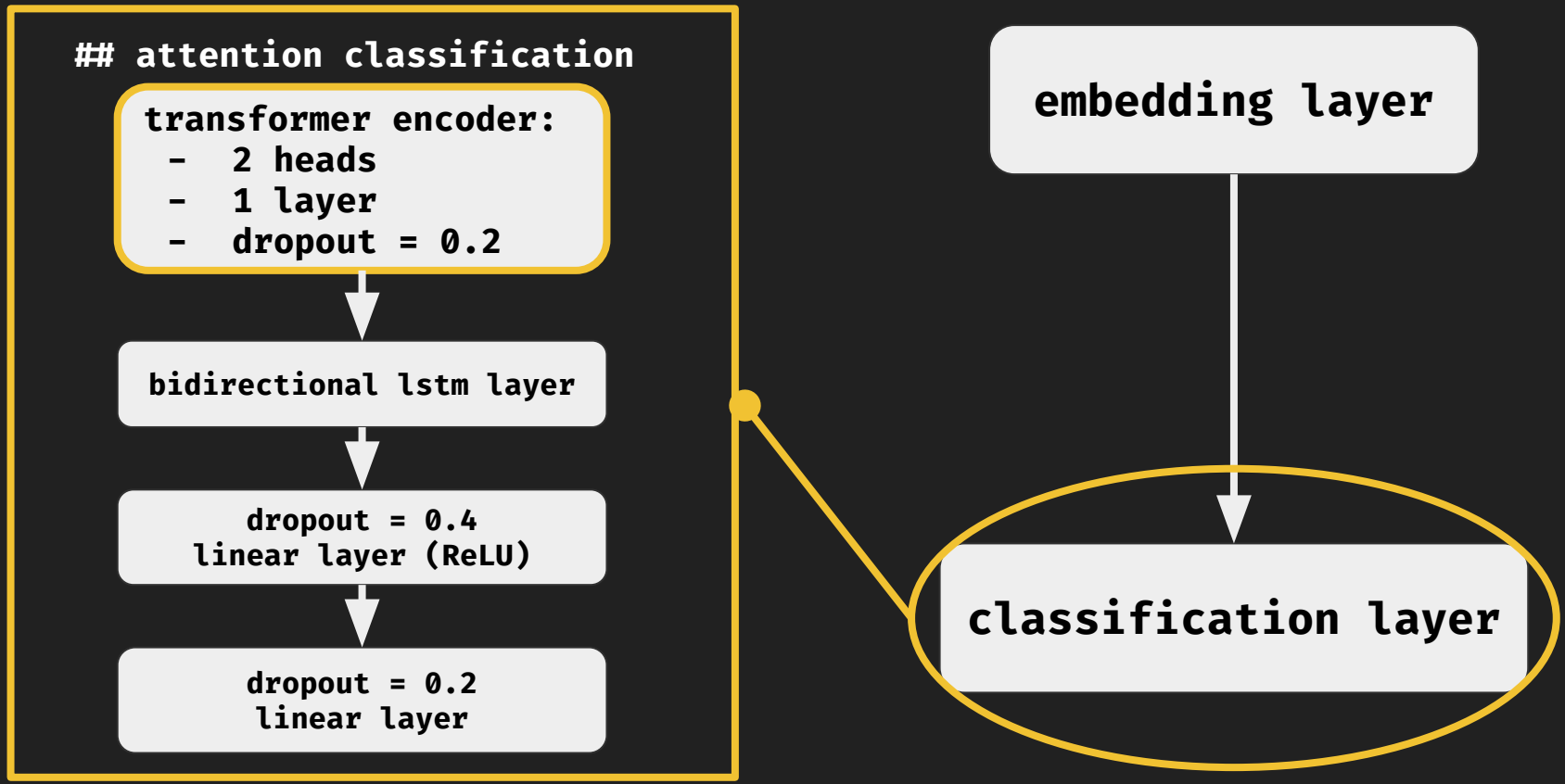
- **cbow**
  - **trained on BookCorpus**
    (embedding size = 100)
  - **pretrained on Google News**
    (embedding size = 300)
- **bert**
  - **bert-tiny**
    (embedding size = 128)
  - **bert-base**
    (embedding size = 768)
  - **distilbert**
    (embedding size = 768)



Training


lstm classification
bidirectional lstm layer
dropout = 0.4 linear layer (ReLU)
dropout = 0.2 linear layer
embedding layer
classification layer

Training


attention classification
transformer encoder:
- 2 heads
- 1 layer
- dropout = 0.2
bidirectional lstm layer
dropout = 0.4
linear layer (ReLU)
dropout = 0.2
linear layer
embedding layer
classification layer

Training

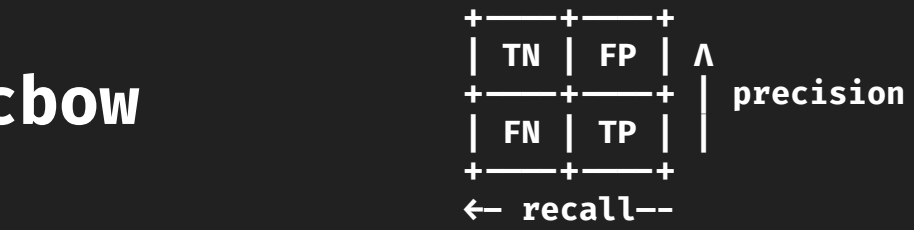
Training settings

setting	loss	optimizer	max # documents	epochs	batch size	sequence length
value	cross entropy	adam	6	5/8	16	128

Class weights for the loss function

classes	[EMPTY]	[PERIOD]	[COMMA]	[QUESTION]
weights	1.0	10.0	20.0	100.0

Evaluation - cbow

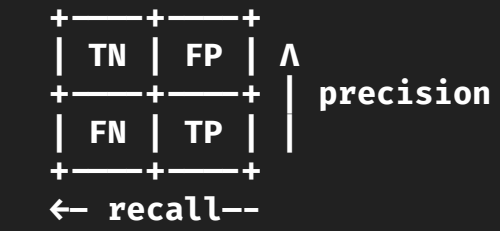


cbow embedding models	[EMPTY]			[PERIOD]			[COMMA]			[QUESTION]			OVERALL W/O EMPTY		
	p	r	f	p	r	f	p	r	f	p	r	f	p	r	f
trained lstm	0.00	0.00	0.00	0.00	0.00	0.00	5.93	92.50	11.14	1.95	22.70	3.60	2.63	38.40	4.91
trained attention	0.00	0.00	0.00	0.00	0.00	0.00	5.72	83.33	10.71	2.72	51.08	5.17	2.82	44.80	5.30
pretrained lstm	0.00	0.00	0.00	0.00	0.00	0.00	5.77	100.0	10.91	0.00	0.00	0.00	1.92	33.33	3.64
pretrained attention	95.90	63.32	76.28	11.33	0.90	1.67	12.27	69.17	20.84	3.86	45.89	7.12	9.15	38.65	9.87

p: precision - r: recall - f: f1

model	trained lstm	trained attention	pretrained lstm	pretrained attention
time per epoch (s)	68	87	78	115

Evaluation - bert



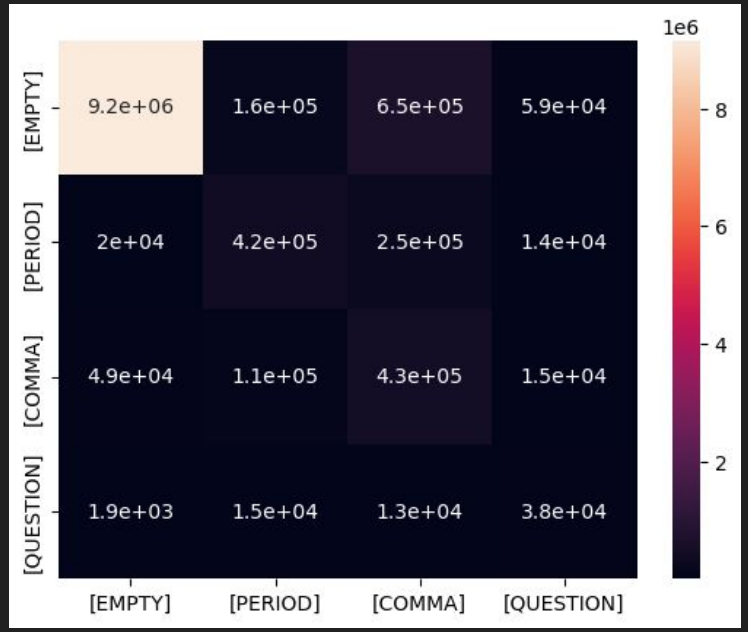
bert embedding models	[EMPTY]			[PERIOD]			[COMMA]			[QUESTION]			OVERALL W/O EMPTY		
	p	r	f	p	r	f	p	r	f	p	r	f	p	r	f
bert-tiny lstm	97.56	72.24	83.01	37.89	8.34	13.68	14.88	65.41	24.25	5.22	57.43	9.57	19.33	43.73	15.83
distilbert lstm	98.93	85.66	91.82	52.87	48.81	50.76	22.18	57.62	32.03	13.66	59.64	22.23	29.57	55.36	35.01
bert-base lstm	99.01	89.75	94.15	59.06	61.90	60.45	28.65	58.81	38.53	21.85	62.73	32.41	36.52	61.15	43.80
bert-base attention	98.99	90.06	94.31	58.89	62.64	60.71	28.83	59.32	38.81	25.26	59.64	35.49	37.66	60.53	45.00

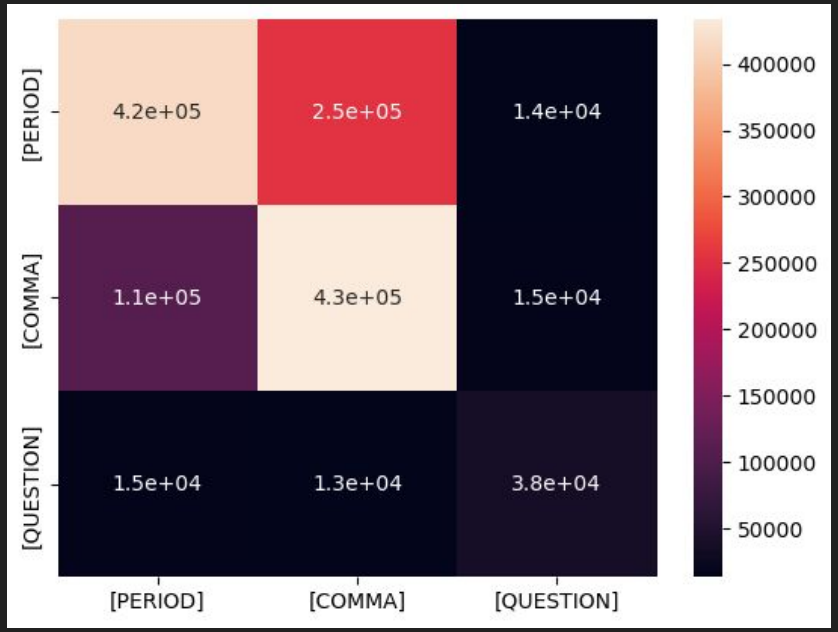
p: precision - r: recall - f: f1

model	bert-tiny lstm	distilbert lstm	bert-base lstm	bert-base attention
time per epoch (s)	93	674	381	742

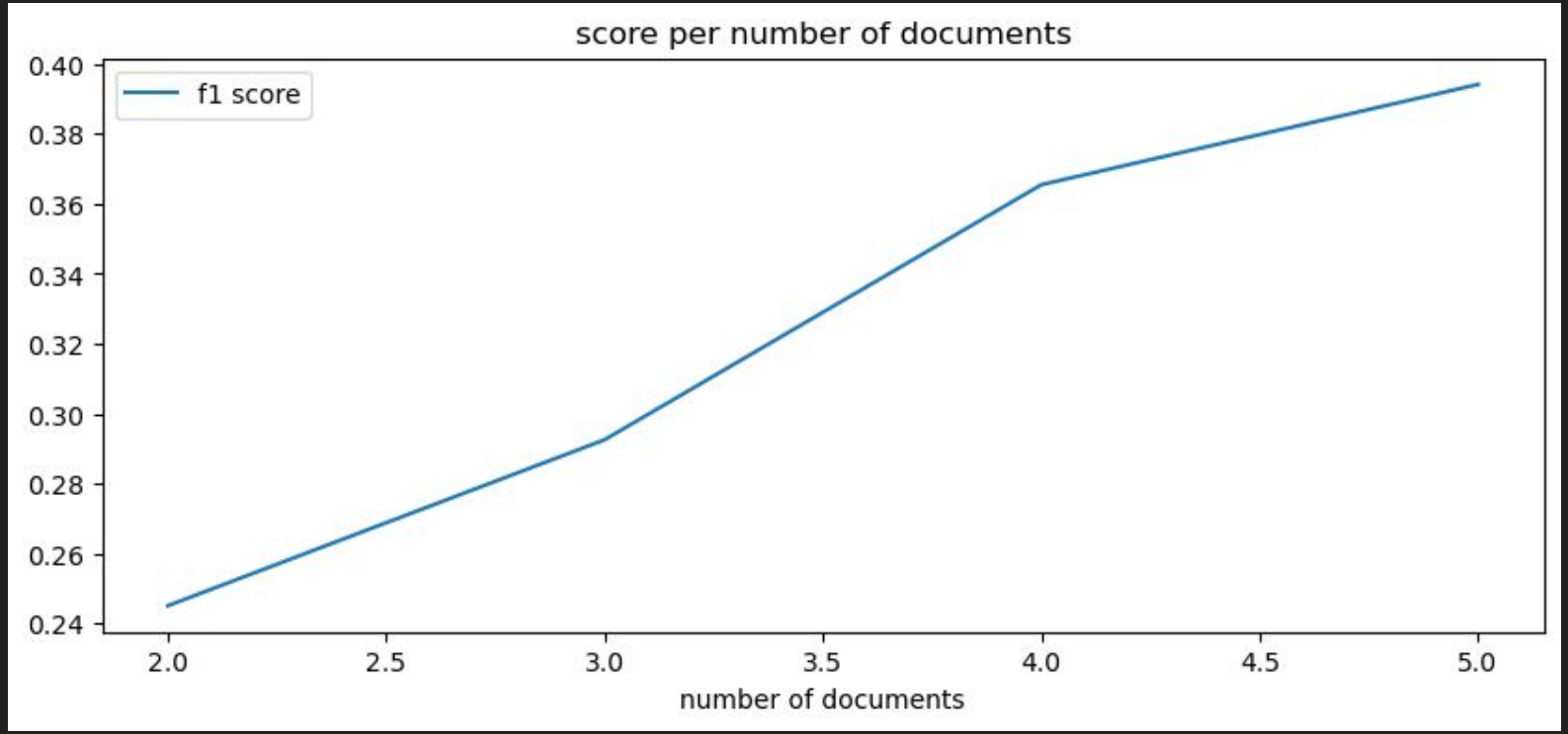
Evaluation - bert-base attention

Confusion matrix





Learning curve - bert-base attention



Prediction - bert-base attention

original text:

... the 102nd and 98th will each contribute two frigates towards filling the gaps in the 51st, 144th and 153rd. those three squadrons, along with the 77th, will be redeployed after their crews stand down for a one week rest period. what ultimately will become of the 102nd and 98th has not been decided yet. we may disband those squadrons altogether, or rebuild them with new ships coming off the shipyards, but thats yet to be determined. now, before i dismiss you so that you can get your crews on the ground, its important that we have all after - action reports before you go on r r. if you havent already ...

predicted punctuation:

... the 102nd and 98th will each contribute two frigates towards filling the gaps in the 51st, 144th and 153rd. those three squadrons along with the 77th will be redeployed after their crews stand down for a one week rest period, what ultimately will become of the 102nd and 98th has not been decided. yet, we may disband those squadrons altogether or rebuild them with new ships coming off the shipyards, but thats yet to be determined. now. before i dismiss you, so that you can get your crews on the ground. its important, that we have all after - action reports before you go on. r. r. if you havent already ...

Conclusion

What we could have done if we had more time

- More data is better, bigger sequence length too
- Adding another prediction head for POS may help predicting punctuation
- Punctuation marks are rich and doing a high level preliminary prediction of the more frequent punctuation is better than predicting less frequent punctuation (e.g. ‘(’, ‘)’, …)

References

- [Punctuation Restoration using Transformer Models for High-and Low-Resource Languages](https://aclanthology.org/2020.wnut-1.18.pdf)
- [Automatic punctuation restoration with BERT models](https://arxiv.org/pdf/2101.07343.pdf)
- [Deep Learning with PyTorch Step-by-Step. A Beginner's Guide. *By Daniel Voigt Godoy*]
- [Hugging Face models and datasets repository](https://huggingface.co/)