max planck institut
informatik

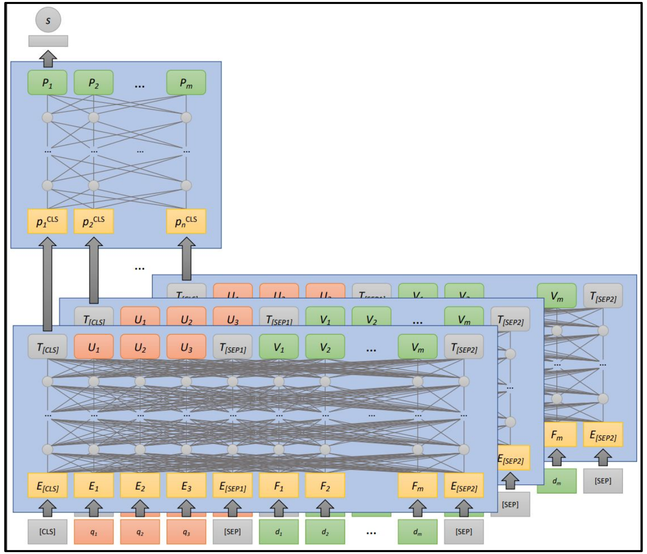
MPII at the NTCIR-15 WWW-3 Task:

Aggregating Passage Representations for Document Reranking

Canjia Li and **Andrew Yates**

Overview

Goal: evaluating passage aggregation strategies (English subtask)



Method: Background

BERT is an effective ranking method, but has limitations

- Inefficient (not addressed here)
- Input cannot exceed 512 tokens

Input length limitation often addressed by aggregating per-passage scores

- Independently compute relevance score for each passage
- Aggregate scores by taking max, first, or sum

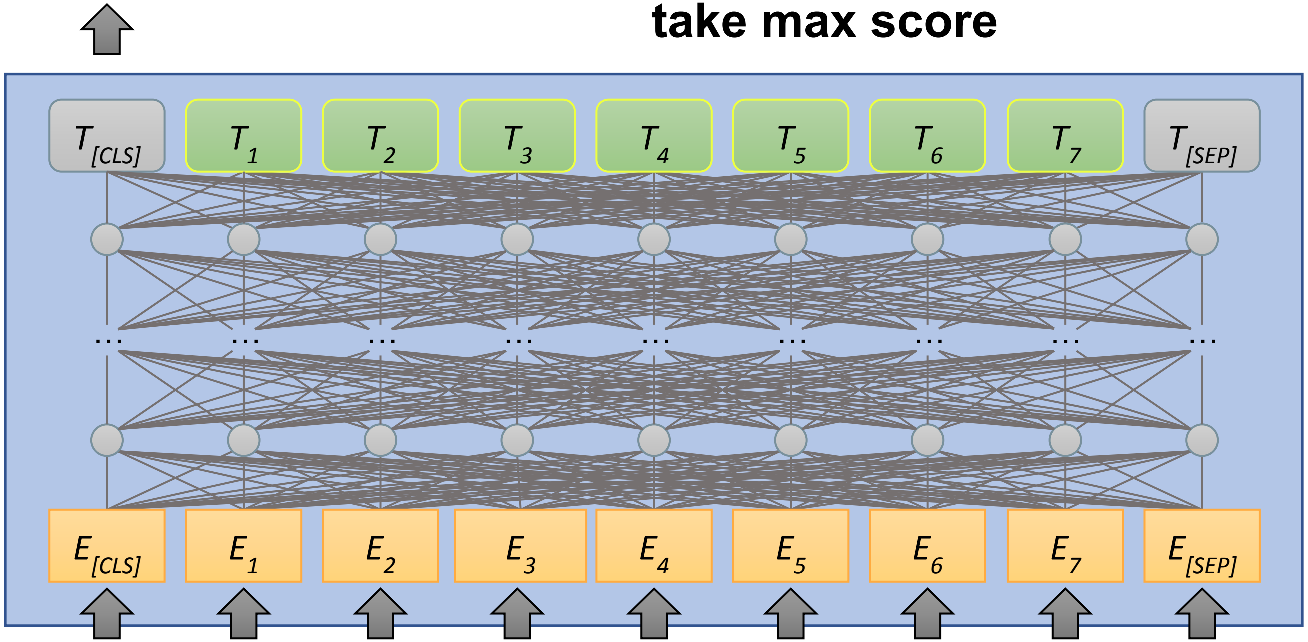
  Deeper Text Understanding for IR with Contextual Neural Language Modeling.
  Zhuyun Dai and Jamie Callan. SIGIR'19.

BERT with Max Scoring Passage (Dai & Callan)

max planck institut
informatik

**score N passages,
take max score**

Passage relevance score



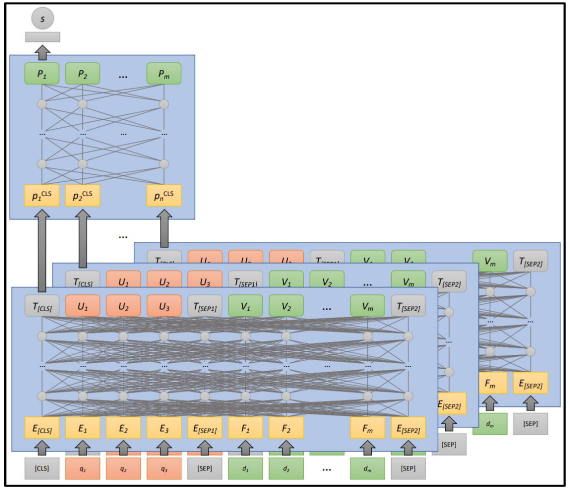
Input: Query (segment A) and Document (segment B)

Method

Prior work suggests MSP not optimal (e.g., Bendersky & Kurland 2008, Fan et al. 2018, Ai et al. 2018)

Idea: aggregate <u>representations</u> rather than <u>scores</u>

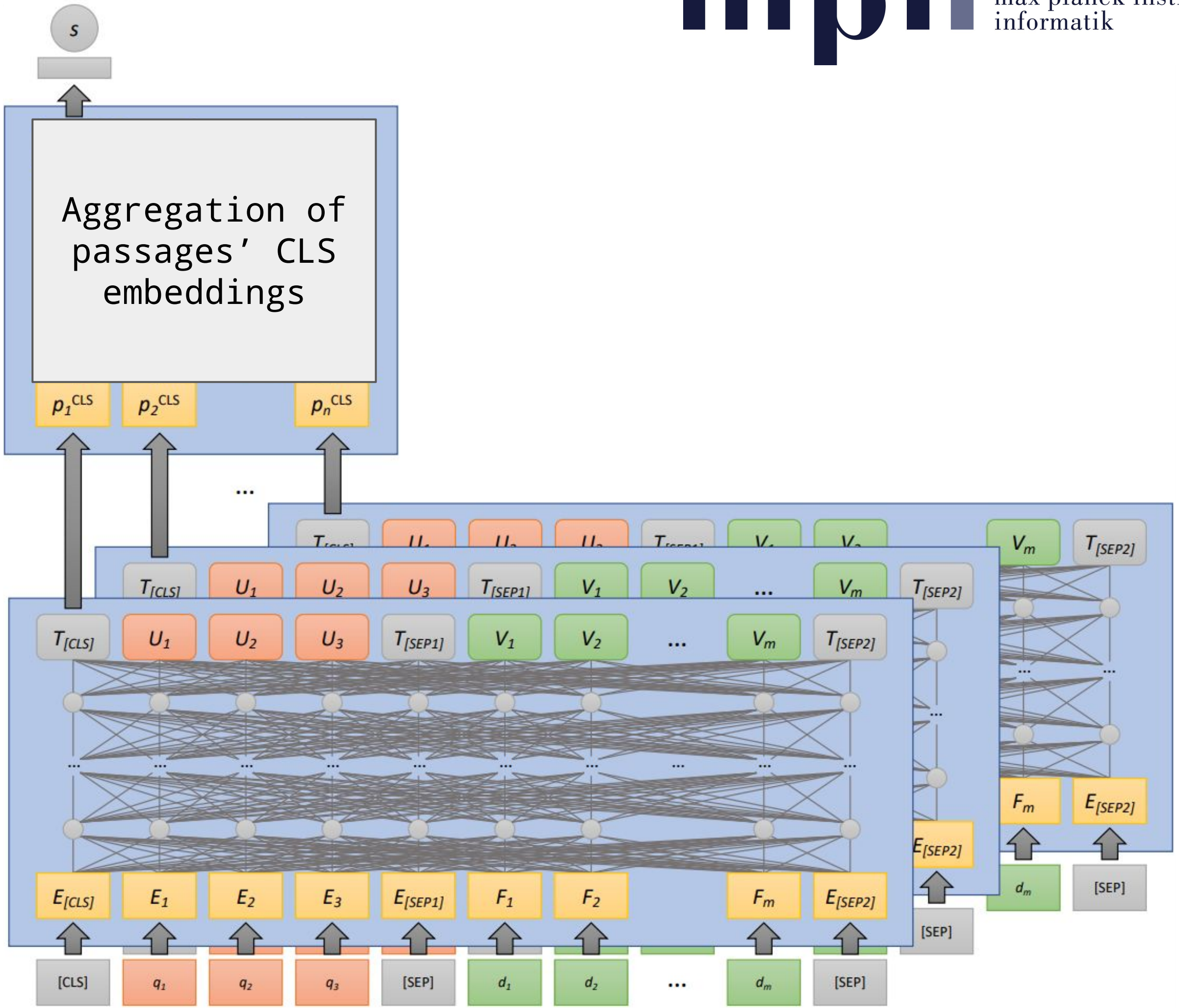
➔ *PARADE: Aggregating Passage Representations for Document Reranking*



PARADE

Aggregation approaches:
(increasing complexity)

- Average feature value
- Max feature value
- Attn-weighted average
- Two Transformer layers



Preprocessing and Training

- Passages: 32 per document of size 150 tokens (stride: 100)
- “BERT” model: ELECTRA-base (Clark et al., ICLR '20) trained on MS MARCO
- PARADE trained on NTCIR WWW-1 and WWW-2
- Rerank top 100 documents from WWW-3 baseline run

Results

Effectiveness increases with aggregation complexity

➔ Transformer > Attn average > Max (all inexpensive relative to BERT)

Aggregations with learned weights much better

➔ {Transformer, Attn} >> Max

Run Name	PARADE variant	nDCG@10	Q@10	nERR@10
mpii-E-CO-NEW-3	$\text{PARADE}_{\text{Max}}$	0.6337	0.6556	0.7395
mpii-E-CO-NEW-2	$\text{PARADE}_{\text{Attn}}$	0.6743	0.6905	0.7787
mpii-E-CO-NEW-1	PARADE	0.6897	0.7016	0.8090

Results

Effectiveness increases with aggregation complexity

➔ Transformer > Attn average > Max (all inexpensive relative to BERT)

Aggregations with learned weights much better

➔ {Transformer, Attn} >> Max

Case study: difficulties with navigational queries

➔ Perhaps a single passage is sufficient here?

160: You want to find the official website of Akron Beacon Journal.

153: You want to visit the website "www.freeweblayouts.net"

Conclusion

Effectiveness increases with aggregation complexity

➔ Mirrors PARADE paper's results on robust04 and GOV2; MS MARCO/DL different

*Canjia Li, **Andrew Yates**, Sean MacAvaney, Ben He, Yingfei Sun. arXiv 2020.*

Overview of BERT-MaxP, PARADE, efficient BERT methods, etc.

➔ *Pretrained Transformers for Text Ranking: BERT and Beyond. Jimmy Lin, Rodrigo Nogueira, **Andrew Yates**. arXiv 2020.*

Implementations of PARADE (& other models): https://capreolus.ai

➔ *Flexible IR Pipelines with Capreolus. Andrew Yates, Kevin Martin Jose, Xinyu Zhang, Jimmy Lin. CIKM '20.*

Conclusion

Effectiveness increases with aggregation complexity

➔ Mirrors PARADE paper's results on robust04 and GOV2; MS MARCO/DL different

*Canjia Li, **Andrew Yates**, Sean MacAvaney, Ben He, Yingfei Sun. arXiv 2020.*

Overview of BERT-MaxP, PARADE, efficient BERT methods, etc.

➔ *Pretrained Transformers for Text Ranking: BERT and Beyond. Jimmy Lin, Rodrigo Nogueira, **Andrew Yates**. arXiv 2020.*

Implementations of PARADE (& other models): https://capreolus.ai

➔ *Flexible IR Pipelines with Capreolus. Andrew Yates, Kevin Martin Jose, Xinyu Zhang, Jimmy Lin. CIKM '20.*

Thanks!
Questions?

Contact:
ayates@mpi-inf.mpg.de
@andrewyates