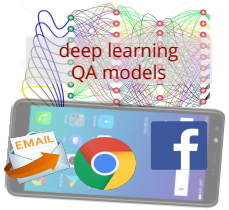
Efficient and Practical Neural Question Answering for Heterogeneous Platforms

Qingqing Cao, joint work with Noah Weber, Harsh Trivedi, Yash Lal, Niranjan Balasubramanian and Aruna Balasubramanian

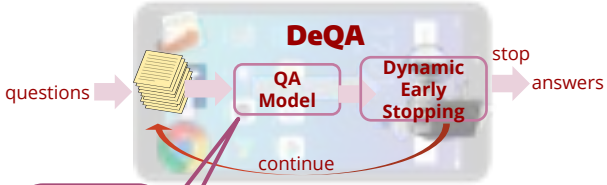
Stony Brook University
Computer Science

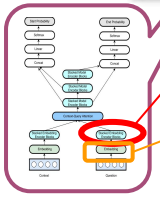
DeQA: On-Device Question Answering (MobiSys 2019)



QA models **don't fit** into a smartphone memory and run **extremely slow**!

Preserve privacy by not sending on-device personal data ✘





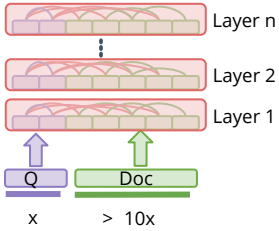
Pre-compute document reps offline to reduce the compute bottleneck

Move memory intensive embeddings off QA model to a key-value store

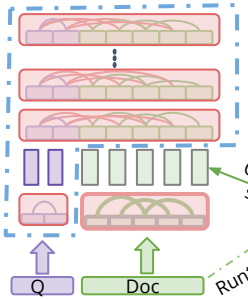
Reduce the QA latency on a smartphone from over a minute to **under 5s**. Reduce energy by >10x.

DeFormer: Decomposing Transformers for Faster Question Answering (ACL 2020)

new challenges for SOTA QA models



- Document is much longer than the question (x > 10x)
- Document processing is the bottleneck in all layers



Run **4x** faster;
Uses **70%** less memory

Minimal accuracy loss (~1% F1 score)

No repeated pre-training

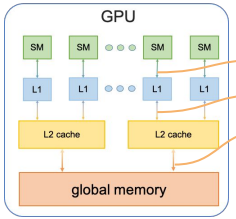
Decomposed Trans**former**

Accurate and Interpretable Energy Modeling for Transformers (under review)

Existing software methods measure the total energy by combining energy of hardware components based on **utilization**

$$e_{total} = PUE \sum_{p} (u_{dram} e_{dram} + u_{cpu} e_{cpu} + u_{gpu} e_{gpu})$$

(Strubell et al., ACL 2019; Henderson et al., JMLR 2020)



Non-utilization behaviors such as **data movement** is the major energy bottleneck (>50%)

(Eyeriss, ISSCC 2016)

Idea: combine runtime resources (include util, freq etc) with model features and feed them into a predictive model to estimate energy