RECOMP: Improving Retrieval-Augmented LMs with Compression and Selective Augmentation

**Key Idea:** Learn a compressor that maps noisy retrieval outputs into a summary and then prepend the summary as an input to LM

![Diagram of the compressor system](image)

---

**Learning a compressor**

**Ingredients**
- Data for downstream task
  - \((q_1, y_1)\), \((q_2, y_2)\), \((q_3, y_3)\)
- Training data

**Step 1:** Generate candidate summaries and evaluate with language model

**Abstractive Compressor**
- Teacher LLM
- \(D\) retrieved documents
- \(q\) query
- \(y\) summary

**Step 2:** Create training data

**Extractive Compressor**
- \(D_1\)
- \(D_2\)
- \(q\) query
- \(y\) summary

**Step 3:** Train the compressor

**Experiments & Results**

- Our method enables efficient inference with minimal performance drop.
- 2x throughput 1/10 tokens!
- Gains from goal-oriented training

% Answer copied

- Top 5 docs
- Ours (ext)
- Ours (abs)
- FLAN-UL2 (20B) results on Natural Questions (left) and TriviaQA (right)

More analysis in the paper!
- Does compressor trained with one LM transferred to other LMs?
- Does compressor trained with one retriever transferred to others?
- How often does selective augmentation happen?
- How good is the teacher model for abstractive compressor?