Summary

- Rag models are better at being truthful than parametric only models
- Parametric memory = seq2seq
- Nonparametric memory = vector index of wikipedia
- Introduces a finetuning method that combines parameter and nonparametric methods

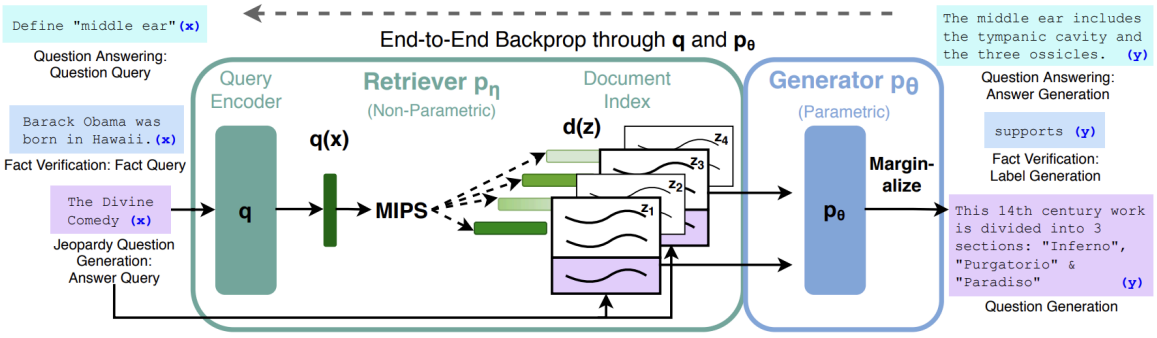
Context

- Pretrained language models can learn a lot from data but cant revise or expand their knowledge, and hallucinate
- Hybrid (ie Rag) models can be updated and could hallucinate less
    - Only need to train the query encoder and generator but **freeze the document index**
    - So when you need to update the corpus, you load it in at inference time
- REALM, ORQA
    - Combines language model with **differentiable retriever**

Parametric vs nonparametric

- In a parametric model, the number of parameters is fixed with respect to the sample size. In a nonparametric model, the (effective) number of parameters can grow with the sample size.
- In other words, a parametric model assumes the form of the data distribution
- A non parametric model does not make assumptions about the underlying data distribution

Methods



- Non parametric part: pre-trained retriever (query encoder plus document index)

        - Dense vector index for wikipedia accessible through a pretrained neural retriever
            - Retriever is known as DPR (dense passage retriever)
        - Also uses topK approximation
- Parametric part: generator
    - Is a pretrained seq2seq transformer

More details from section 2

- Uses input x to retrieve text documents z_1…z_4
    - Text documents are for extra context
- The non parametric retriever
- Used additional context for y
- Retrieval augmented generation is the fine tuning method

Rag sequence

- Uses a singular retrieved document to generate complete sequence
- The retrieved doc is used as a single latent variable

Rag token

- Allows generator to **select content from multiple documents as context**
- top K documents are retrieved using the retriever, and then the generator produces a distribution for the next output token for each document

Dpr: retriever

- The part that retrieves documents
- Is a bi-encoder structure, similar to bert?

Generator: bart

- Uses BART-large

Training

- Both retriever and generator are trained

- Minimizes the negative marginal log likelihood of each target

$$\sum_j -\log p(y_j|x_j)$$

    - Using stochastic gradient descent with adam

Decoding

- Decoding is done at test time
- Rag sequence and rag token do different things here
- Rag token: standard, autoregressive seq2seq generator with transition probability
- Rag seq:
    - cannot solve it with a single beam search

Experiments

- They use the december 2018 wikipedia dump for their nonparametric model
    - They use a document encoder after splitting it into 100 word chunks
    - Then use MIPS (Maximum inner-product search) probably to search for most relevant documents relating the input query
    - Then return the top K documents for each query
- They test on these tasks in the following sections:

Open domain question answering

- asks a model to produce answers to factoid questions in natural language.

Abstractive question answering

- Questions that are not common knowledge and that require searching of sources

Jeopardy question answering

- Input: “In 1986 Mexico scored as the first country to host this international sports competition twice.”
- Output: “The World Cup”

Fact verification

- classifying whether a natural language claim is supported or refuted by Wikipedia, or whether there is not enough information to decide

Related works [COOL]

-

Closing statements

- the fact that it is more strongly grounded in real factual knowledge (in this case Wikipedia) makes it “hallucinate” less with generations that are more factual, and offers more control and interpretability
- Downside: what if the nonparametric model dataset is not accurate? Ie we cant guarantee wiki is completely accurate

Questions

- “top K documents are retrieved using the retriever, and then the generator produces a distribution for the next output token for each document”
    -
- “Updating the document encoder BERTd during training is costly as it requires the document index to be periodically updated as REALM does during pre-training [20]. We do not find this step necessary for strong performance, and keep the document encoder (and index) fixed, only fine-tuning the query encoder BERTq and the BART generator.”
    - Wat does this mean
- What do they mean by decoding? I assume its the forward pass after the model has been trained of the generator cuz generator is referred to as the decoder right?
- What is marginalizing
    - Refers to selection of documents

    RAG-Sequence Model The RAG-Sequence model uses the same retrieved document to generate the complete *sequence*. Technically, it treats the retrieved document as a single latent variable that is marginalized to get the seq2seq probability $p(y|x)$ via a top-K approximation. Concretely, the top K documents are retrieved using the retriever, and the generator produces the output sequence probability for each document, which are then marginalized,

    $$p_{\text{RAG-Sequence}}(y|x) \approx \sum_{z \in \text{top-}k(p(\cdot|x))} p_\eta(z|x) p_\theta(y|x,z) = \sum_{z \in \text{top-}k(p(\cdot|x))} p_\eta(z|x) \prod_i^N p_\theta(y_i|x,z,y_{1:i-1})$$

    RAG-Token Model In the RAG-Token model we can draw a different latent document for each target *token* and marginalize accordingly. This allows the generator to choose content from several documents when producing an answer. Concretely, the top K documents are retrieved using the retriever, and then the generator produces a distribution for the next output token for each document, before marginalizing, and repeating the process with the following output token, Formally, we define:

    $$p_{\text{RAG-Token}}(y|x) \approx \prod_i^N \sum_{z \in \text{top-}k(p(\cdot|x))} p_\eta(z|x) p_\theta(y_i|x,z,y_{1:i-1})$$

-

    - What does this mean
-