Empower Language Model with Knowledge in ODQA

reviewed by Lee Huhyun (sltask0222@gmail.com)

(Expressions in this document were initially taken from the referenced original papers.)

Open-Domain Question Answering (ODQA) is the task of answering general domain questions, in which the evidence is not given as input to the system. Perhaps it is one of the most knowledge-intensive tasks in natural language processing.

Large scale language models showed they can extract factual information by training on vast quantities of data (Radford et al., 2019; Petroni et al., 2019; Jiang et al., 2019; Talmor et al., 2019).

- ✓ Radford et al. (2018) showed **GPT-2**, which is a transformer-based pre-trained language model with sufficient capacity (1.5 billion parameters) can perform down-stream tasks including QA in a zero-shot setting without any parameter or architecture modification.
- ✓ Radford et al. (2020) trained a 175 billion parameter autoregressive language model, which called **GPT-3** and showed that the 'in-context learning' abilities of language model with high capacity transformer architecture can be improved by growing up its scale.
- ✓ **BERT** (Devlin et al., 2019) uses masked language models to enable pre-trained deep bidirectional representations. It differs from GPT-2 (Radford et al., 2018), and GPT-3 (Radford et al., 2020), which uses unidirectional language models for pre-training, and also differs from ELMo (Peters et al., 2018), which uses a shallow concatenation of independently trained left-to-right and right-to-left LMs.
- ✓ **T5** (Rafael et al., 2020) is a unified framework that converts all text-based language problems into a text-to-text format. It can be viewed as an application of transfer learning for variant NLP tasks, where during the transfer learning, a model is first pre-trained on a data-rich task before being finetuned on a downstream task.
- ✓ Pre-trained neural language models learn a substantial amount of in-depth knowledge from data without any access to an external memory (as a parametrized implicit knowledge base). But they do have downsides: cannot easily expand or revise their memories (storage space is limited by the size of the network to capture more world knowledge, so one must train ever-larger networks, which can be prohibitively slow or expensive), cannot straightforwardly provide insight into their predictions, and may produce "hallucinations".

Most ODQA works assume the model can access an external text corpus so they retrieve relevant passages as knowledge, and thus named **Open-Book models** (Roberts et al., 2020)

- ✓ Chen et al. (2017) tackled open-domain question answering by combining the challenges of document retrieval (finding the relevant articles) with that of machine comprehension of text (identifying the answer spans from those articles). They used Wikipedia as the unique knowledge source to answer any factoid question.
- ✓ **REALM** (Guu et al., 2020) combine masked language models with a differentiable retriever. This approach explicitly exposes the role of world knowledge by asking the model to decide what knowledge to retrieve and use during inference. First before making each prediction, the language model uses the retriever to retrieve documents from a large corpus such as Wikipedia, and then attends over those documents to help inform and "extract" its prediction. REALM selects its best documents with Maximum Inner Product Search (MIPS) algorithm.

- ✓ **DPR** (Karpukhin et al., 2020) uses BERT as dense encoders to leverage dense vector representations for retrieval, creating a vector space such that relevant pairs of questions and passages will have smaller distance (i.e., higher similarity) than the irrelevant ones. Encoded passages are indexed with FAISS. The dense, latent semantic encoding helps retrieving synonymous contexts, unlike traditional sparse vector methods such as TF-IDF or BM25.

- ✓ **BART** (Lewis et al., 2020) is a denoising autoencoder for pretraining sequence-to-sequence models that maps a corrupted document to the original document where it was derived from. BART shows effectiveness not only for text generation but also for comprehension tasks.

- ✓ **RAG** (Lewis et al., 2020) is a hybrid model which combines parametric memory with non-parametric (i.e., retrieval-based), for knowledge-intensive NLP tasks. RAG combines DPR as a retriever (called a non-parametric memory, can be seen as a large external memory for neural networks to attend to) with BART as a generator (called a parametric memory). Both memory components are pre-trained and pre-loaded with extensive knowledge so they can access knowledge without additional training. A key feature of RAG is that it is comprised of raw text rather distributed representations, which makes the memory both (i) human-readable, lending a form of interpretability to our model, and (ii) human-writable, enabling us to dynamically update the model's memory by editing the document index. RAG is more strongly grounded in real factual knowledge, makes it "hallucinate" less with generations that are more factual, and offers more control and interpretability.

- ✓ Izacard and Grave (2020) also leverages both retrieval and generative modeling for ODQA, first retrieving supporting passages using either sparse or dense representations (BM25 or DPR) and then generating the answer by a sequence-to-sequence model (T5 or BART), taking as input the retrieved passages in addition to the question. This approach is analoguos to RAG, but differs from it in those works by how the generative model processes the retrieved passages.

Another line of ODQA works assume knowledge could be stored implicitly in parameters of Language Models (LMs) and thus named **Closed-Book models** (Roberts et al., 2020).

- ✓ **Roberts et al.** (2020) introduced a generative model for open domain question answering. Without relying on external knowledge, this method obtained competitive results on several benchmarks. However, it requires models containing billions of parameters, since all the information needs to be stored in the weights. This makes models expensive to query and train.

- ✓ Closed-Book models generate answers without retrieving from an external corpus and thus benefit from faster inference speed and simpler training. However, This setting is much harder as it requires LM to memorize all pertinent knowledge in its parameters, and even recent LMs with much larger model parameters is still not competitive to state-of-the-art Open-book models.

To improve the knowledge coverage of LM, one natural choice is to leverage knowledge stored in **Knowledge Graph (KG)**.

- ✓ **PullNet** (Sun et al., 2019) uses an iterative process to construct a question-specific subgraph that contains information relevant to the question. In each iteration, a graph convolutional network (graph CNN) is used to identify subgraph nodes that should be expanded using retrieval (or "pull") operations on the corpus and/or Knowledge Base (KB). After the subgraph is complete, another graph CNN is used to extract the answer from the subgraph. So PullNet iteratively expands the subgraph by choosing nodes from which to "pull" information about, from the KB or corpus as appropriate.

- ✓ **ERNIE** (Zhang et al., 2019) explicitly inject knowledge representation into language model pre-

training, encoding the KGs with knowledge embedding algorithms like TransE (Bordes et al., 2013) as input. Based on the alignments between text and KGs, ERNIE integrates entity representations in the knowledge module into the underlying layers of the semantic module.

- ✓ **JAKET** (Yu et al., 2022) implicitly model knowledge information into language model by performing entity category prediction, jointly pre-training both the KG representation and language representation. The knowledge module produces embeddings for entities in text and the language module generates context-aware initial embeddings for entities and relations in the graph. They provide essential information to mutually assist each other.
- ✓ **UnifiedSKG** (Xie et al., 2022) unifies 21 Structured knowledge grounding (SKG) tasks into a text-to-text format, aiming to promote systematic SKG research, instead of being exclusive to a single task, domain, or dataset. The task families include Semantic Parsing, Question Answering, Data-to-Text, Conversational, Fact Verification and Formal-Language-to-Text. Different from unifying tasks that only take text as input such as T5, a core challenge in unifying SKG tasks into the text-to-text format is to linearize structured knowledge. They convert the input x into an input sequence by unifying user request with linearized structured knowledge and the context. As structured knowledge, tables and highlighted tables, relation-triples, knowledge graph and ontology were linearized into a sequence.