Open Domain Question Answering

New approaches, new possibilities

Chris Falter, Indiana U Data Science Program

https://github.com/chrisfalter/DataScience

Open Domain Question Answering: What we will cover

- Background
- Methods
- Dense Passage Retrieval (DPR)
- DPR Results
- Technical Challenges
- Future Directions
- Using DPR Outside the Lab
- References

Open Domain Question Answering: Background

“The task of answering questions using a large collection of documents of diversified topics” [Chen, ACL2020]

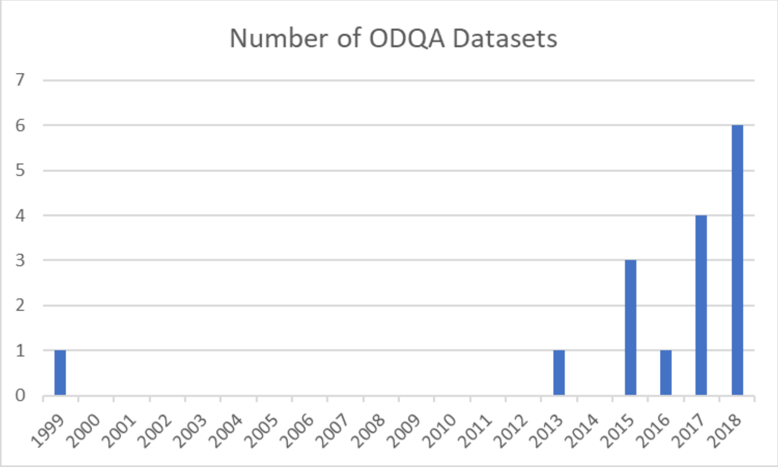
ODQA is a challenging task comprised of several sub-tasks:

- Natural language understanding
- Information retrieval
  - Search
  - Ranking
- Expressing the answer
  - Extraction
  - Natural language generation

An example of *extractive* question answering

ODQA Background

- Flavors of ODQA:
  - Text-based (using Wikipedia, medical publications, etc.)
  - Knowledge graph-based (using Freebase, WikiData)
- Interest has been exploding
  - Large number of benchmark data sets have become available
  - Funding and activity from key technical, government, and academic institutions



ODQA Methods

1960s: Early question answering could not exceed limited domains

```
Month = July
  Place = Boston
    Day              = 7
    Game Serial No. = 96
    (Team = Red Sox, Score = 5)
    (Team = Yankees, Score = 3)
```

BASEBALL, 1961 [Green]

Photo credit: https://www-03.ibm.com/press/us/en/photo/33488.wss

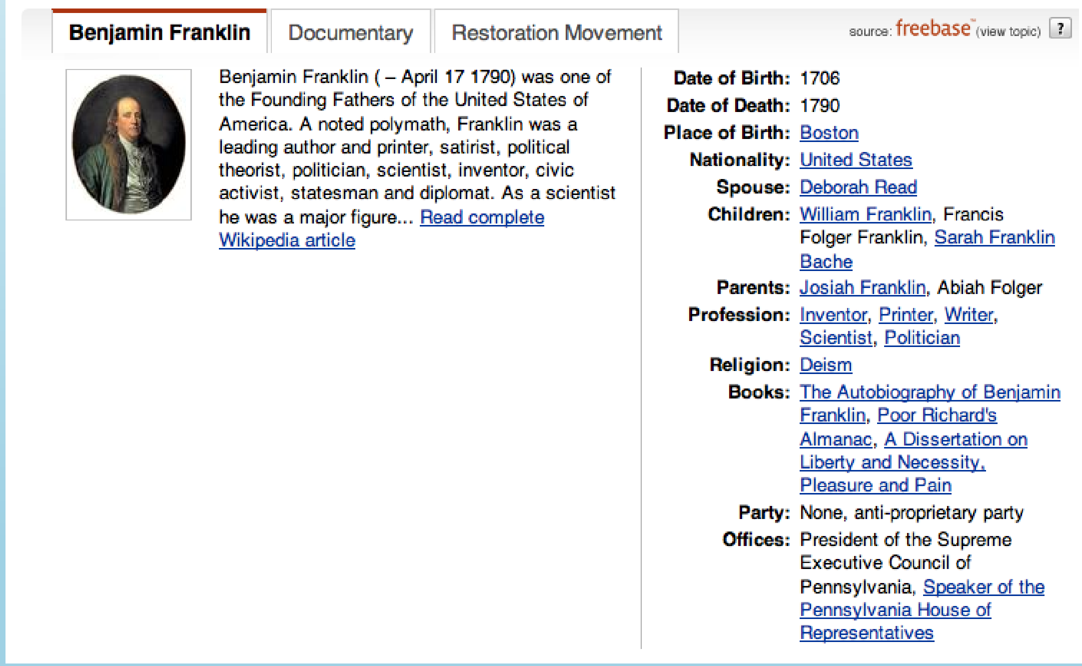
1999 – 2012: Statistical models rule

- Relied on information redundancy to select most likely response [Chen, ACL2020]
- Required deep engineering resources
  - IBM Watson [Ferrucci, 2012]
    - Dozens of engineers, scientists, and linguists
    - Several years of effort
    - Inference: 900 servers, 2880 cores

ODQA Methods

2013 – present: Answering from structured knowledge bases

- Capable of useful results, BUT …
- Needs a large ontology to provide semantic parsing of data, and
- Fact-gathering requires enormous human curation
- Thus the method is
  - Limited in range
  - Unable to use semantic similarity [Chen, 2017]

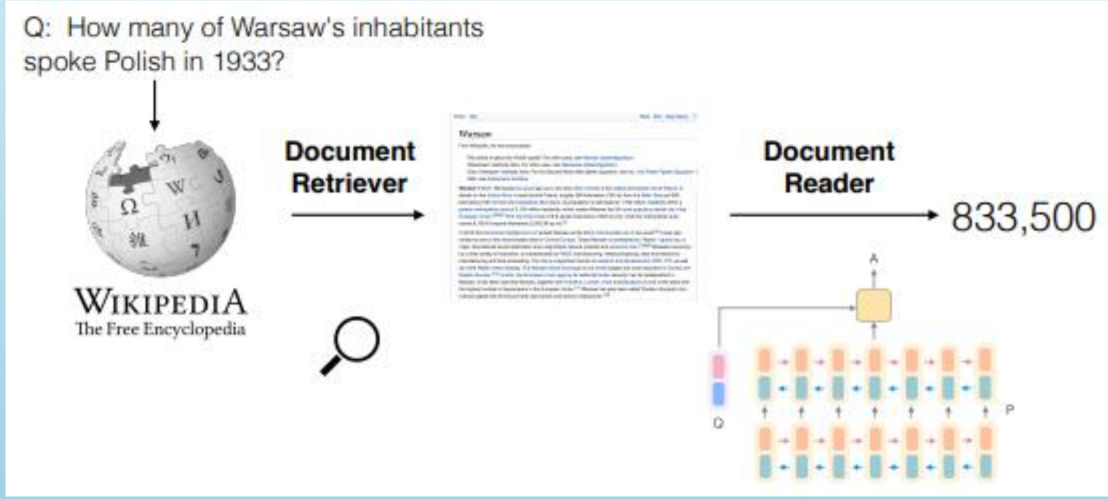


ODQA Methods

2017 – present

Two-stage retriever-reader

- Neural reader can use semantic similarities of words and sentences
- Limitations [Karpukhin, 2020]
  - Retriever cannot use semantic similarities
  - Retriever cannot be trained



[Chen, 2017]

Dense Passage Retrieval

Dense Passage Retrieval for Open-Domain Question Answering

Vladimir Karpukhin[*], **Barlas Oğuz**[*], **Sewon Min**[†], **Patrick Lewis,**
Ledell Wu, Sergey Edunov, Danqi Chen[‡], **Wen-tau Yih**
Facebook AI [†]University of Washington [‡]Princeton University
{vladk, barlaso, plewis, ledell, edunov, scottyih}@fb.com
sewon@cs.washington.edu
danqic@cs.princeton.edu

Abstract

Open-domain question answering relies on efficient passage retrieval to select candidate contexts, where traditional sparse vector space models, such as TF-IDF or BM25, are the de facto method. In this work, we show that retrieval can be practically implemented using *dense* representations alone, where embeddings are learned from a small number of questions and passages by a simple dual-encoder framework. When evaluated on a

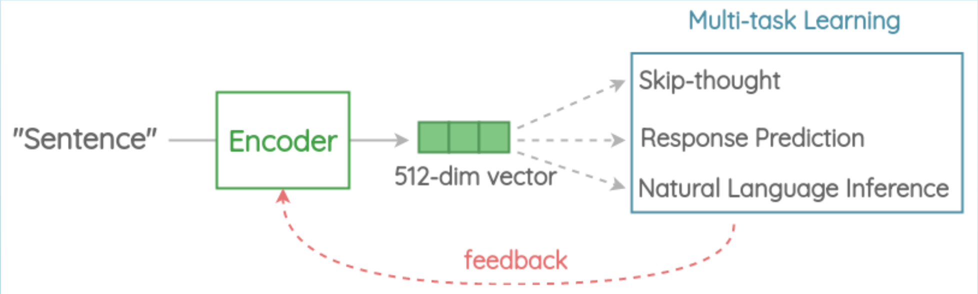
Retrieval in open-domain QA is usually implemented using TF-IDF or BM25 (Robertson and Zaragoza, 2009), which matches keywords efficiently with an inverted index and can be seen as representing the question and context in high-dimensional, sparse vectors (with weighting). Conversely, the *dense*, latent semantic encoding is *complementary* to sparse representations by design. For example, synonyms or paraphrases that consist of completely different tokens may still be mapped to

- Uses passage embeddings to semantically match an embedded query to the most responsive passages
- Still a two-stage retriever-reader pipeline. However,
  - The retriever uses semantic similarity
  - The reader uses the embedded query to identify the answer text from retrieved passages

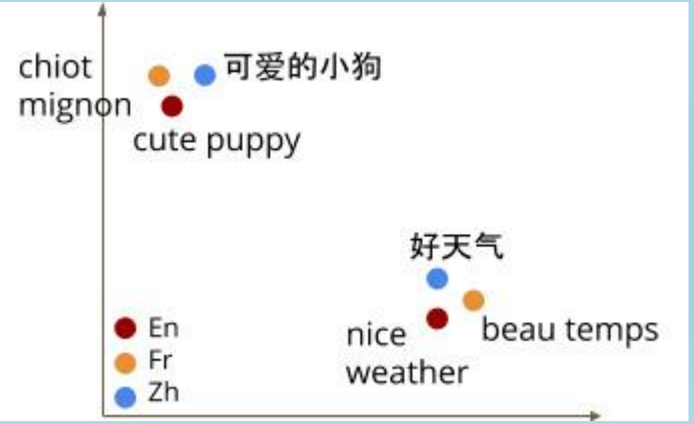
Why DPR Uses Passage Embeddings

Passage embedding = vector representation (encoding) of a passage

Contains semantic content (similarities and differences in word and sentence meanings) by virtue of training methods



[Chaudhury, 2020]



[Yang, 2020]

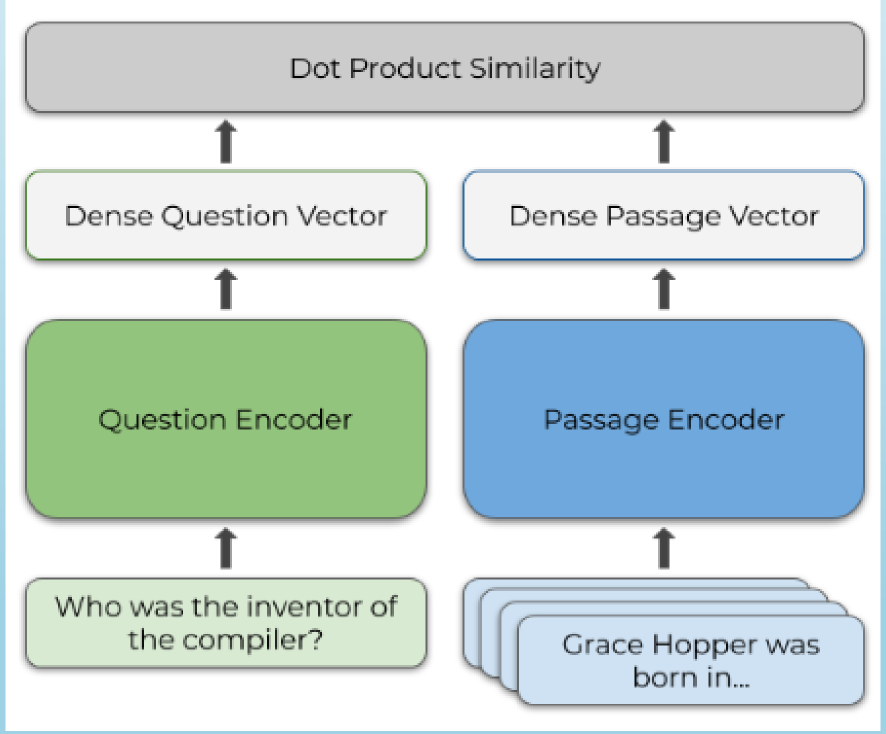
The semantic capabilities of passage embeddings are illustrated by Google's (multi-language) universal sentence encoder

How DPR Uses Passage Embeddings

Intuition: Dot product measures the semantic similarity of question and passage

Model Training: jointly train question encoder and passage encoder to maximize dot product similarity of responsive passages and minimize dot product similarity of non-responsive passages

Inference: Select *k* passages whose embeddings are most similar to question embedding



[May, 2020]

DPR Results

- DPR achieves SOTA on 4/5 benchmarks
- Performs poorly when trained on small datasets.
  - Must be trained on more data (multiple datasets) to achieve good metrics
- Performs better on real questions harvested from server logs (WebQuestions, NaturalQuestions) than on questions formulated when answer is already known (TriviaQA, SQuAD) [Chen, 2020]

Training	Model	NQ	TriviaQA	WQ	TREC	SQuAD
Single	BM25+BERT (Lee et al., 2019)	26.5	47.1	17.7	21.3	33.2
Single	ORQA (Lee et al., 2019)	33.3	45.0	36.4	30.1	20.2
Single	HardEM (Min et al., 2019a)	28.1	50.9	-	-	-
Single	GraphRetriever (Min et al., 2019b)	34.5	56.0	36.4	-	-
Single	PathRetriever (Asai et al., 2020)	32.6	-	-	-	**56.5**
Single	$\text{REALM}_{\text{Wiki}}$ (Guu et al., 2020)	39.2	-	40.2	46.8	-
Single	$\text{REALM}_{\text{News}}$ (Guu et al., 2020)	40.4	-	40.7	42.9	-
	BM25	32.6	52.4	29.9	24.9	38.1
Single	DPR	**41.5**	56.8	34.6	25.9	29.8
	BM25+DPR	39.0	57.0	35.2	28.0	36.7
Multi	DPR	**41.5**	56.8	**42.4**	49.4	24.1

DPR Technical Challenges

- Training language models is difficult and expensive
  - Start with pre-trained model (BERT)
  - Incorporate negative samples as well as positive

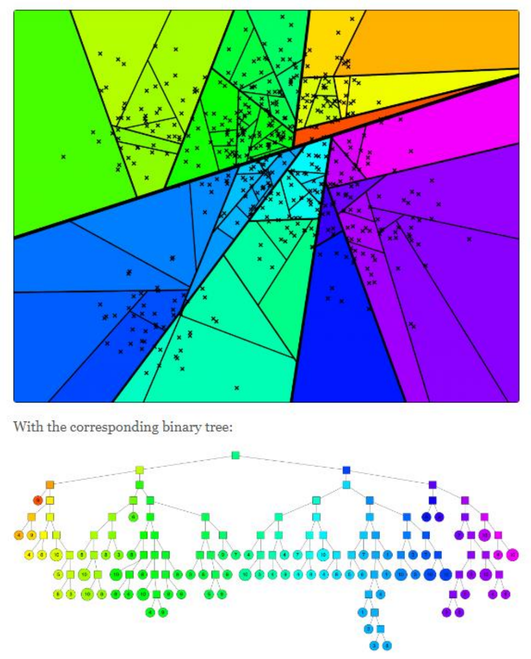
Positives

(1) Provided in the reading comprehension datasets
(2) Passages of high BM25 scores that contain the answer string

Negatives

(1) Random passages from the corpus
(2) Passages of high BM25 scores that DO NOT contain the answer string
(3) Positive passages of OTHER questions

- Nearest neighbors is hard with millions of vectors
  - Use approximate nearest neighbors



[Bernhardsson, 2015]

Future Directions

- Improved training of query models [Xiong, 2020]
- Augment dense passage retrieval with statistical retrieval methods (BM25)
- Use natural language generation (NLG) for use in a dialog system. [Lewis, 2020]

Using DPR Outside the Lab

- Dialog-based search with semantic capabilities can be very helpful for organizations with large case files.
  - Translating thought to keyword search doesn’t always work
    - Imposes cognitive burden on analyst
    - Misses semantically matching passages
- Idea: Experiment by pairing a DPR system with an existing keyword-based search system to form a single search UX
  - The two approaches have complementary strengths

References

Bernhardsson, Erik. 2015. Nearest neighbors and vector models – part 2 – algorithms and data structures. https://erikbern.com/2015/10/01/nearest-neighbors-and-vector-models-part-2-how-to-search-in-high-dimensional-spaces.html

Chaudhury, Amit. 2020. Universal Sentence Encoder Visually Explained. https://amitness.com/2020/06/universal-sentence-encoder/

Chen, Danqi, et al. 2017. Reading Wikipedia to Answer Open-Domain Questions. https://arxiv.org/pdf/1704.00051v2.pdf

Chen, Danqi. 2020. ACL2020 Tutorial: Open-Domain Question Answering. https://github.com/danqi/acl2020-openqa-tutorial

Ferrucci, D. A. 2012. Introduction to "This is Watson." https://researcher.watson.ibm.com/researcher/files/us-heq/W(3)%20INTRODUCTION%2006177724.pdf

Green, Bert. 1961. Baseball: An Automatic Question-Answerer. https://web.stanford.edu/class/linguist289/p219-green.pdf

Karpukhin, Vladimir, et al. 2020. Dense Passage Retrieval for Open-Domain Question Answering. https://arxiv.org/abs/2004.04906

Lewis, Patrick, et al. 2020. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems (NeurIPS)*.

May, Madison. 2020. Representation Learning and Retrieval. https://www.pragmatic.ml/language-modeling-and-retrieval/

Xiong, Wenchin, et al. 2020. Progressively pretrained dense corpus index for open-domain question answering. https://arxiv.org/abs/2005.00038