Neural Search:
Scaling QA to Multiple Documents

Polish NLP Group, 23 July 2020

Branden Chan

Research

deepset

Industry

Surge in Deep Learning based NLP

- Transfer Learning
- Language Models
- Question Answering (QA)

Haystack: Industrial Neural Search

- Driven by QA
- Open Source
- Enterprise features

SIEMENS

AIRBUS

hawk:AI

DATEV

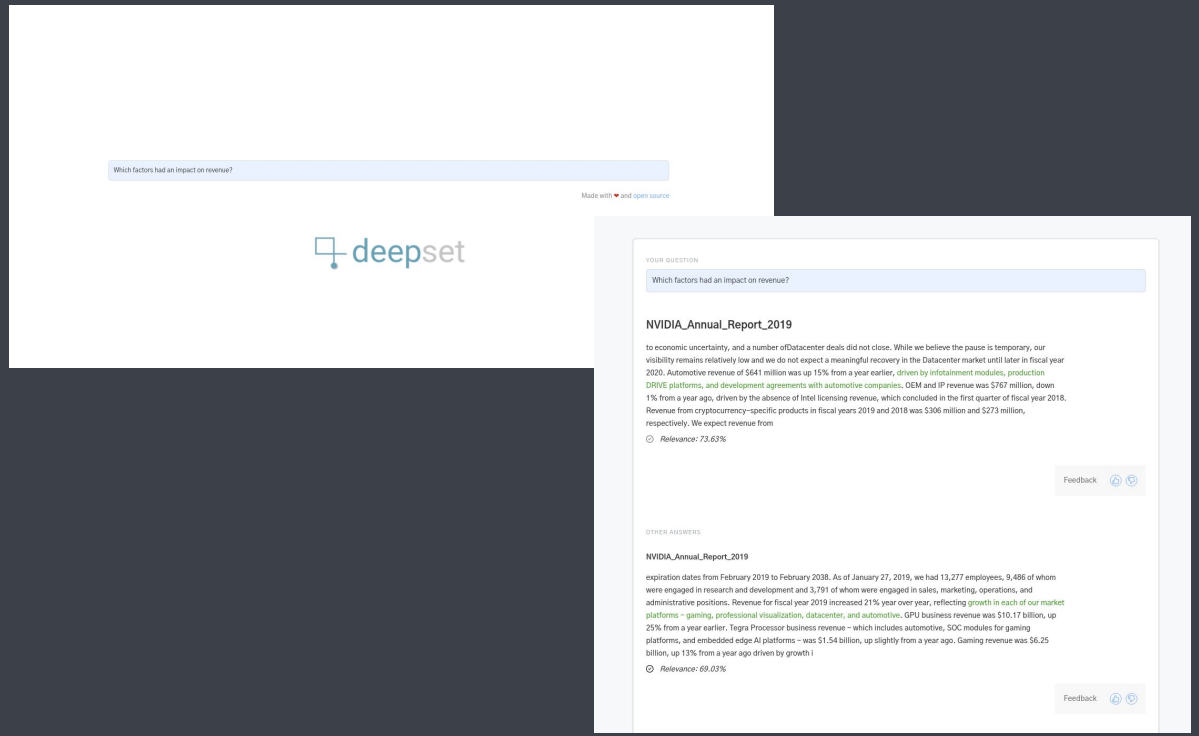
infineon

e·on

SPRINGER NATURE

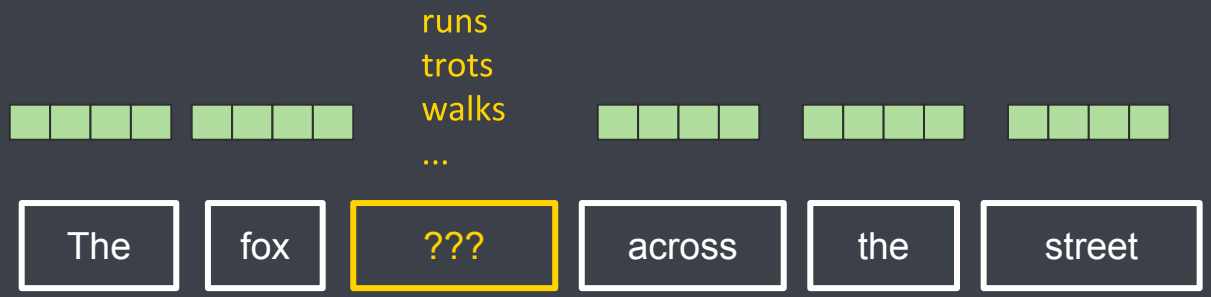
pwc

Demo: Haystack



Language Models (LMs)

What?



How?

Required Data	→	Large Unlabelled Text Corpora
Pretraining Tasks	→	Masked LM + Next Sentence Prediction
Downstream Tasks	→	Exchange prediction heads and fine-tune

Examples

BERT | RoBERTa | XLNET | MiniLM

The language model family

ELMO

ULMFit

Transformer

BERT

Bi-directional LM

GPT

Masked spans

SpanBERT

+Data
+Train time
+Batchsize
-NSP

RoBERTa

Shared weights
NSP ⟶ Sentence order
Factorized Embedding

ALBERT

-Model size

Q-BERT

DistilBERT

MiniLM

The language model family

ELMO

ULMFit

Transformer

BERT

Bi-directional LM

+Data
+Model size

GPT

GPT-2

GPT-3

++ Model size

Defense

Grover

Permutation LM
TransformerXL
+Data

XLNet

XLM

Crosslingual

MT-DNN

Multitask

+Knowledge graph

ERNIE (Tsinghua)

-Model size

Q-BERT

DistilBERT

MiniLM

Text2Text
+Data

T5

Masked spans

SpanBERT

+Data
+Train time
+Batchsize
-NSP

RoBERTa

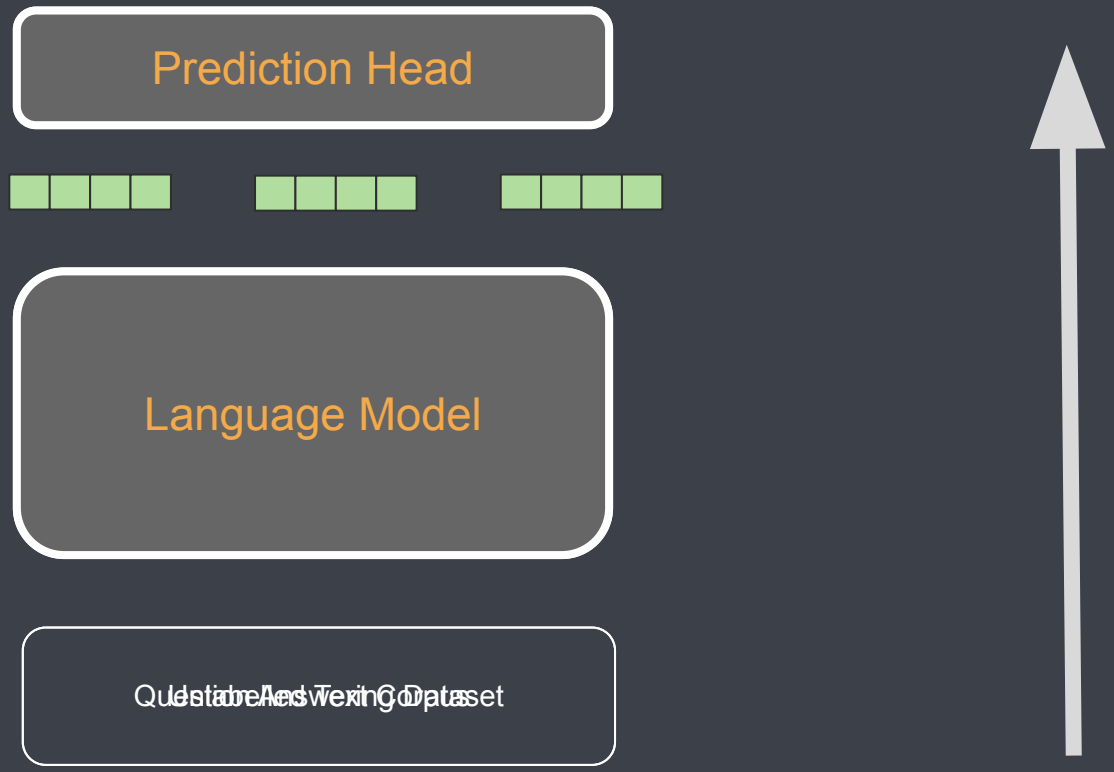
Shared weights
NSP → Sentence order
Factorized Embedding

ALBERT

Replaced token detection

ELECTRA

Modern NLP Workflow using Transfer Learning



Extractive Question Answering: Find answer span in one passage

Question:

What is Berlin?

One text passage:

Berlin (/bɜːrˈlɪn/, German: [bɛʁˈliːn]) is the largest city of Germany by both area and population. Its 3,748,148 (2018)[2] inhabitants make it the second most populous city proper of the European Union after London.The city is one of Germany's 16 federal states. It is surrounded by the state of Brandenburg, and contiguous with Potsdam, Brandenburg's capital. The two cities are at the center of the Berlin-Brandenburg capital region, which is, with about six million inhabitants and an area of more than 30,000 km²,[4] Germany's third-largest metropolitan region after the Rhine-Ruhr and Rhine-Main regions [...]

Answer:

the largest city of Germany

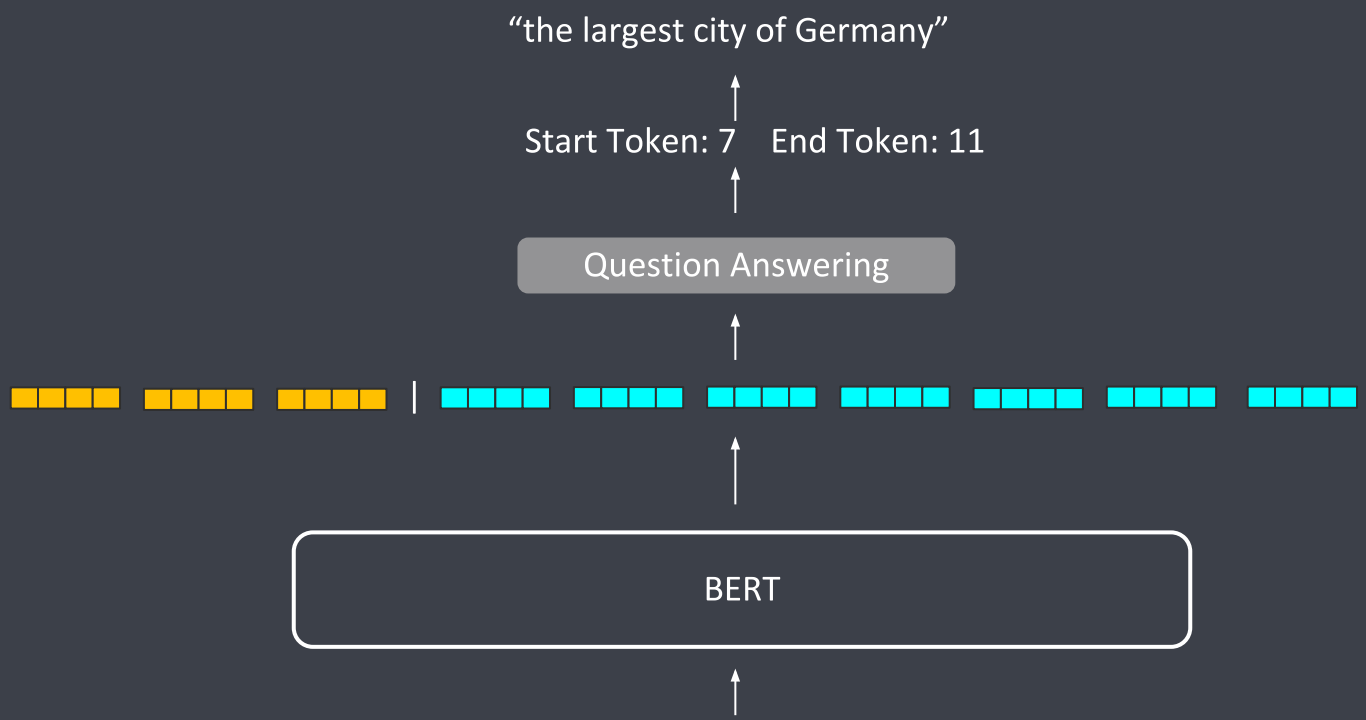
Challenges

- Interaction of text and question
- Large set of potential predictions
- Contextual understanding of potentially long text

Popular approach

- Train “Reader” model to identify start and end of answer span
- Fine-tune a language model on labelled question-answer-pairs
- SQuAD has become a default dataset

Question Answering: Modelling of the Reader

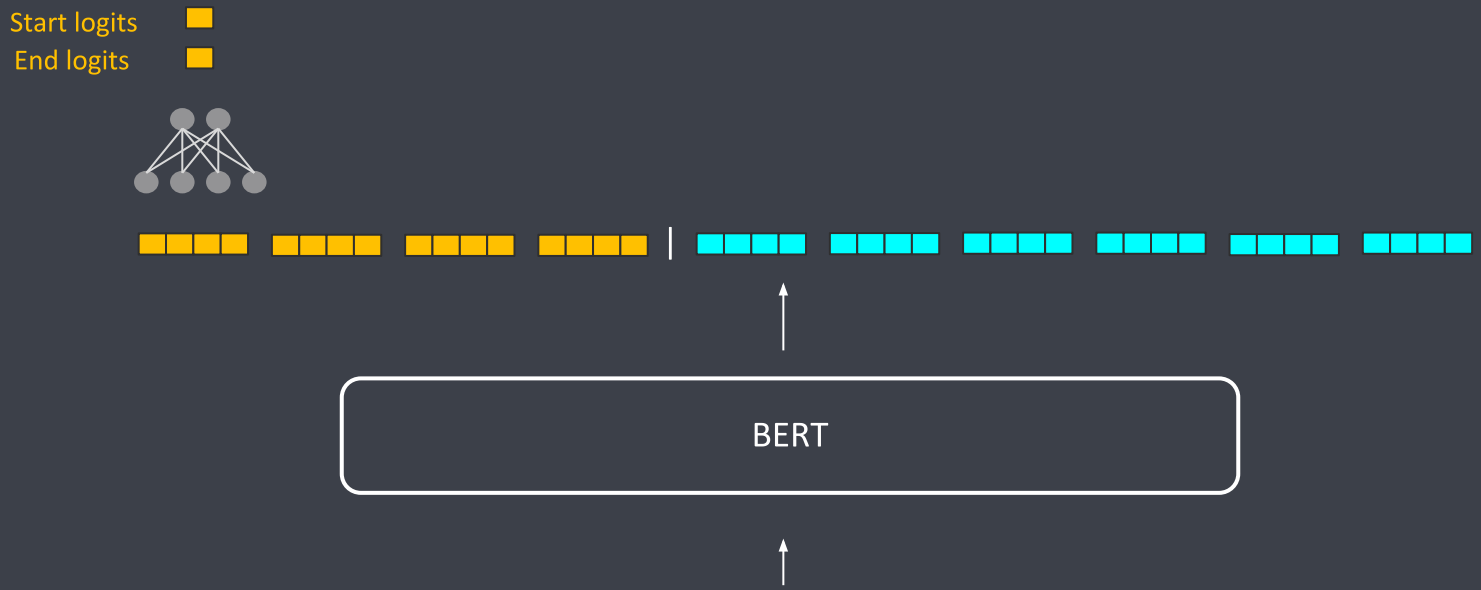


Prediction Head

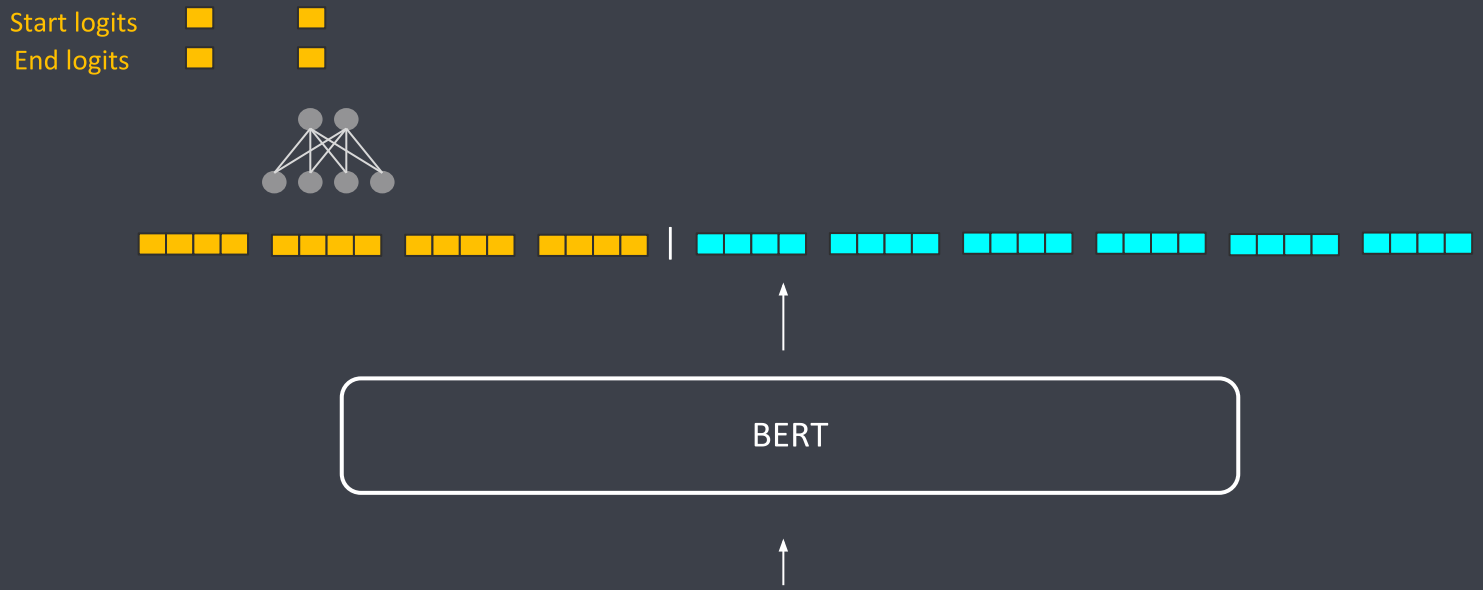
Contextualized Token Embeddings

Language Model

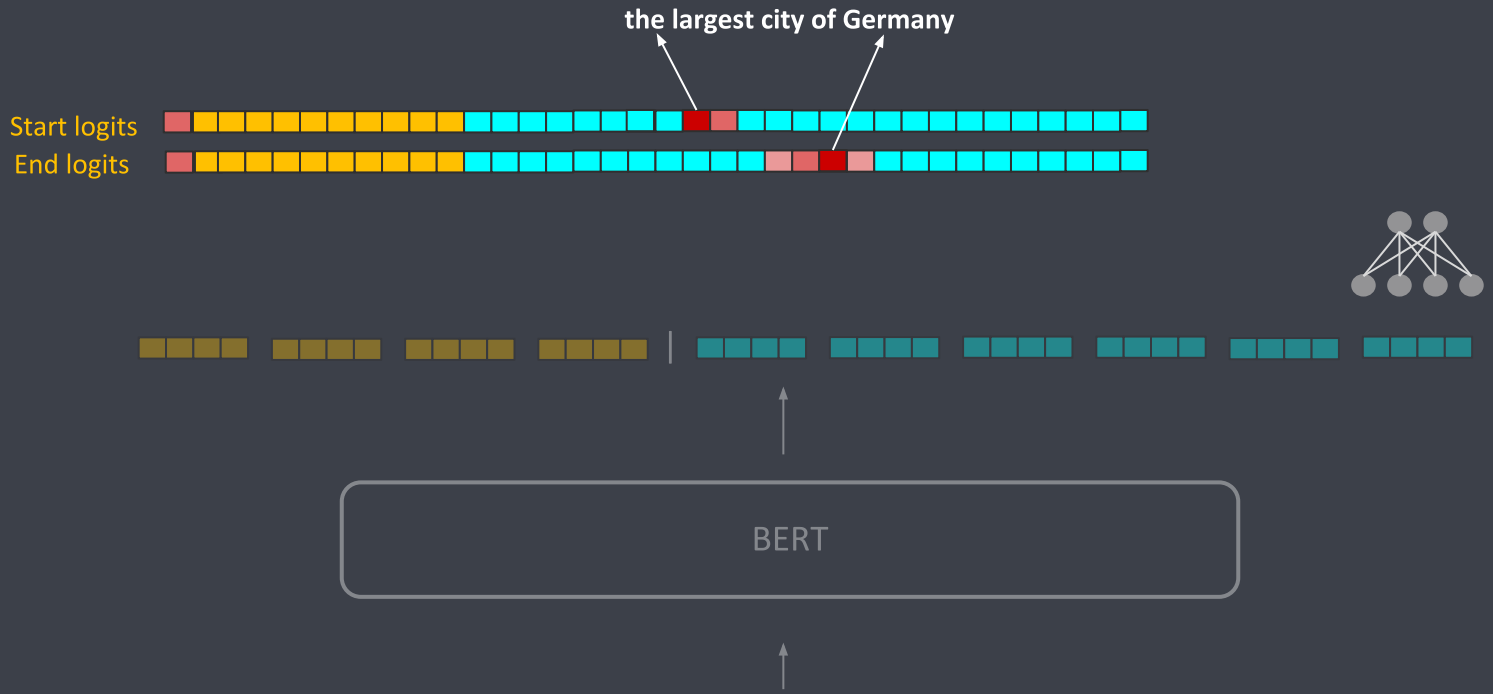
Question Answering Modelling - Details



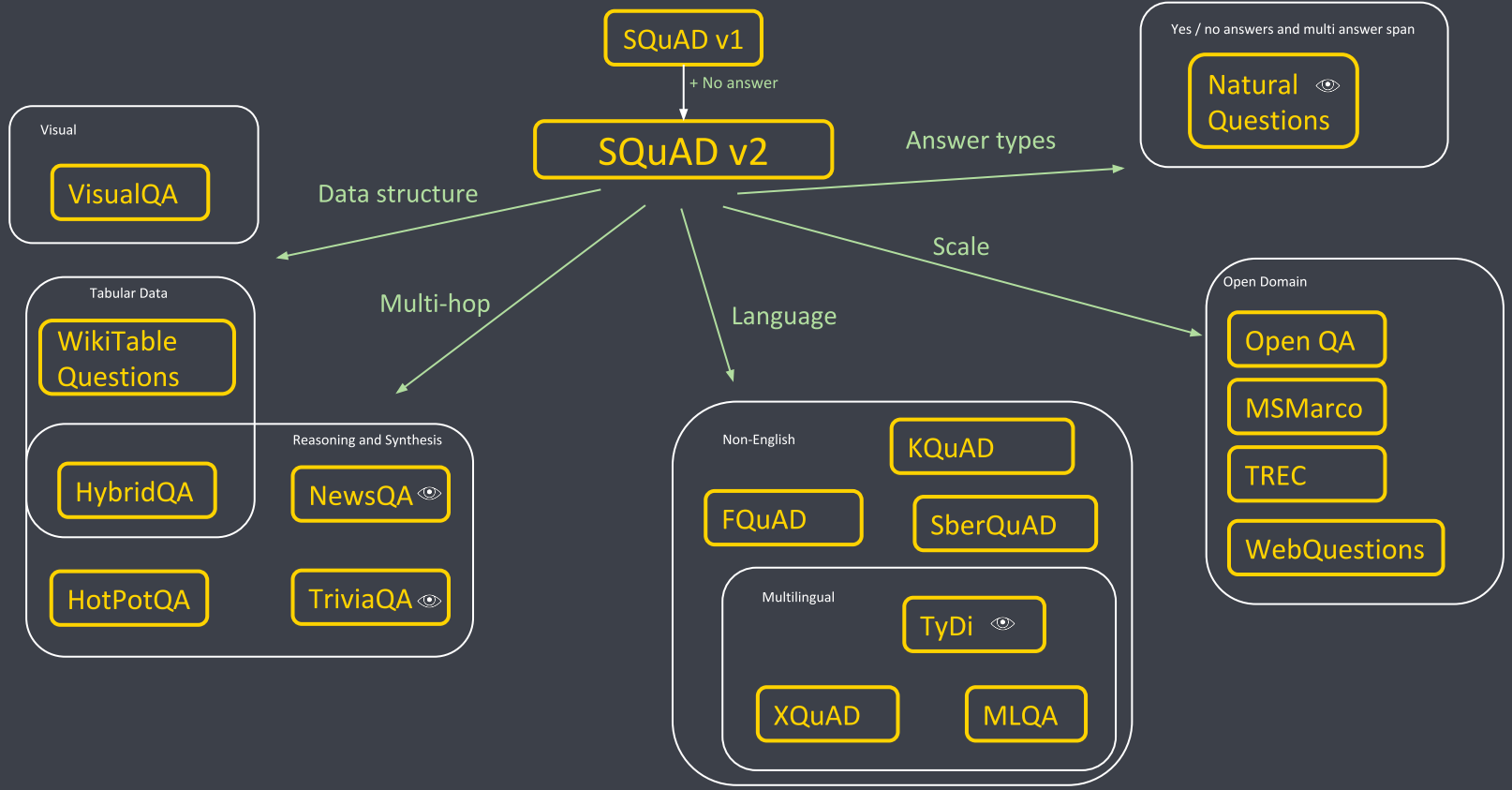
Question Answering Modelling - Details



Question Answering Modelling - Details



Overview of QA Datasets

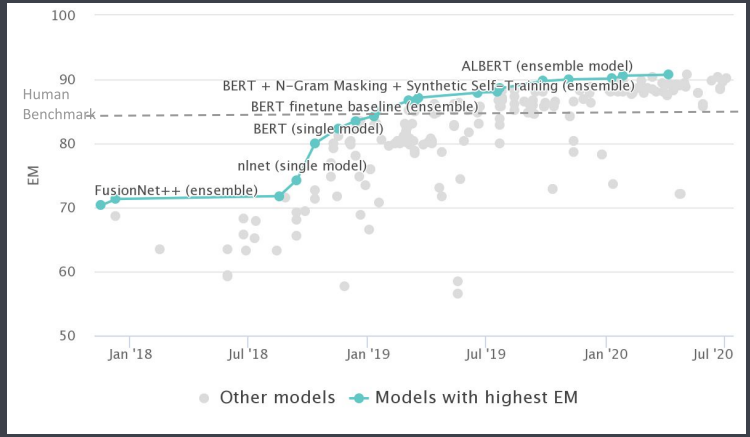


Question Answering: Why now?

- Transformer models boosted accuracy
  and enabled practical usage

- Big momentum in research
  - Relevance: New models, Reasoning & Synthesis
  - Speed: Distillation, Pruning, Sparse Attention
  - Other languages

- High demand from industry
  - Growing number of “information workers”
  - Growing gap between web search and enterprise search
  - Use cases beyond search



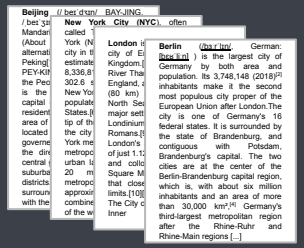
Source: paperswithcode.com/sota/question-answering-on-squad20

Open-Domain QA: Find answer span in large collection of documents

Question:

What is Berlin?

Large collection of documents:



Answer:

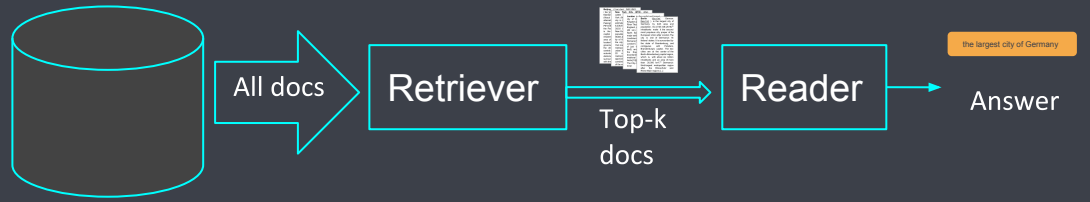
the largest city of Germany

Challenges

- Limited input length for reader models
- Speed
- Aggregation of predictions

Popular approach

- Retriever-Reader-Pipeline

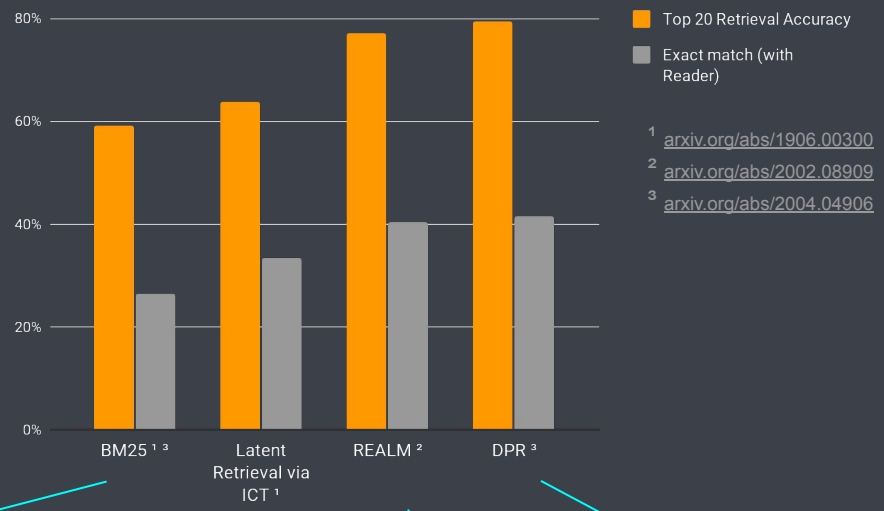


Reader: 3+ hours / query
w/ Retriever: 0.5-2 sec / query

(7.5k docs from NQ dev, RoBERTa-base, Tesla V100)

State of the Art Components

State of the art retrievers



[1] arxiv.org/abs/1906.00300
[2] arxiv.org/abs/2002.08909
[3] arxiv.org/abs/2004.04906

Sparse baseline (e.g. Elasticsearch)

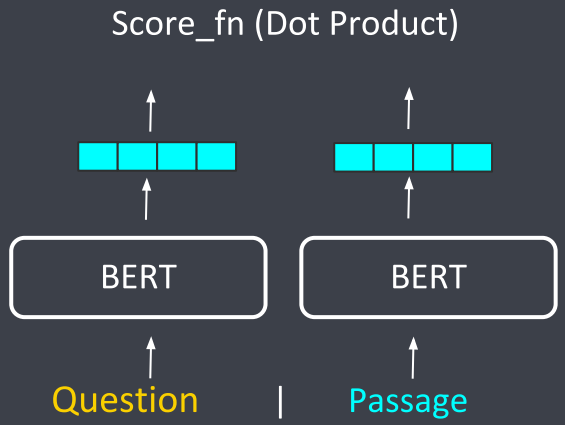
Training dual encoders via Inverse Cloze task on raw text

Retrieval augmented language model pretraining

Training dual encoders on question-passage pairs with in-batch negatives

DPR: “Dense Passage Retrieval for Open-Domain QA”

Dual Encoders



Training

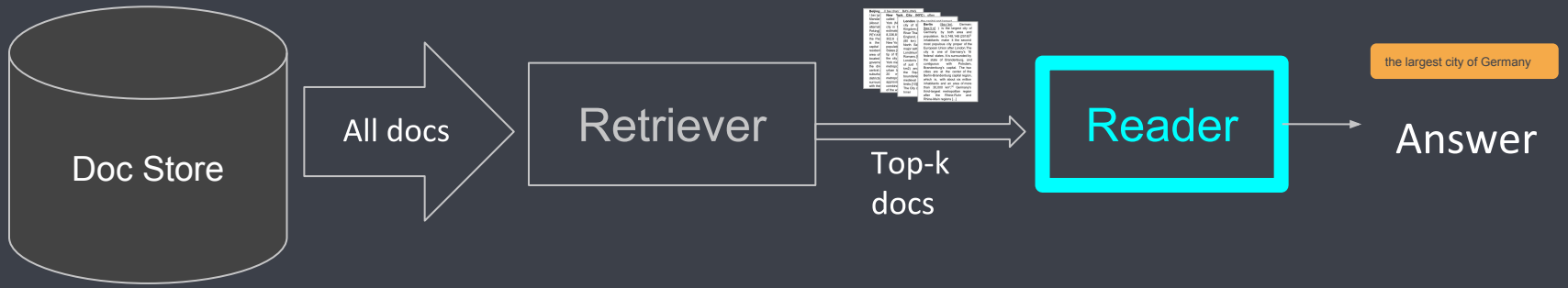
- Sample: 1x question, 1x pos. passage, n x neg. passages
- Negative Sampling: In-batch negatives + 1x BM25
- Combining multiple QA datasets helps

Inference

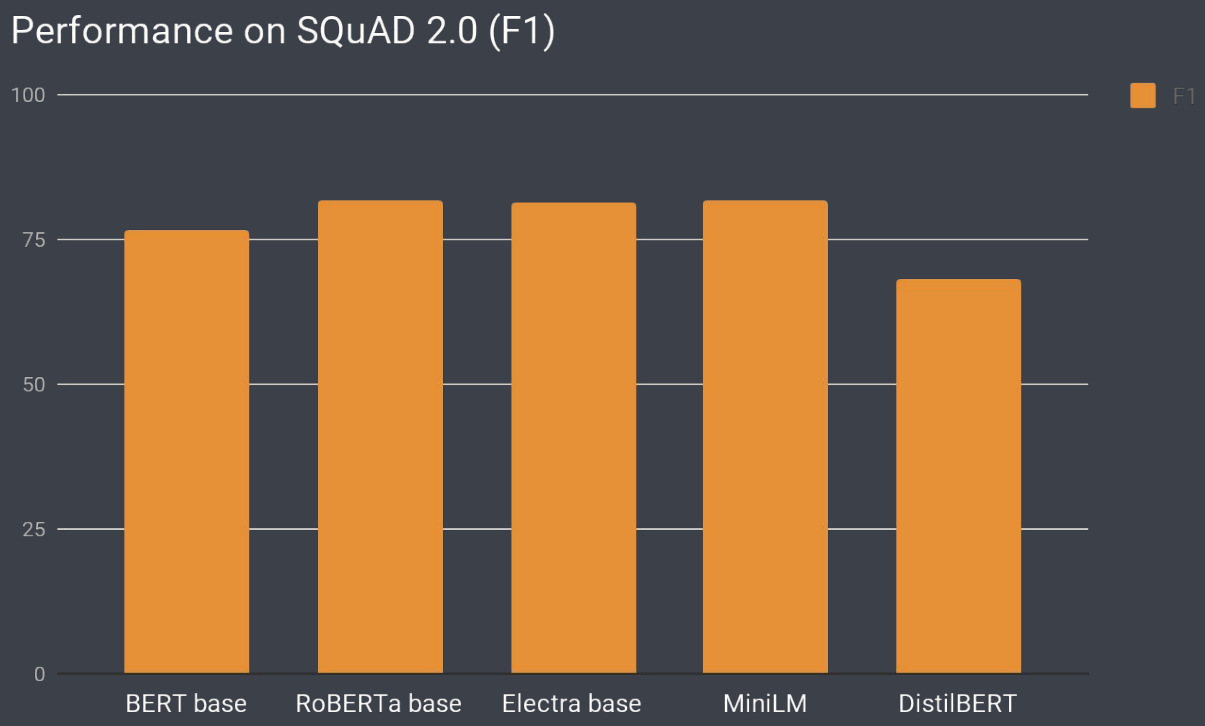
- Pre-compute embeddings and index to FAISS
- Retrieve k nearest neighbours

Karphukin et al. (2020): https://arxiv.org/abs/2004.04906

State of the Art Components



What you can do on the reader side



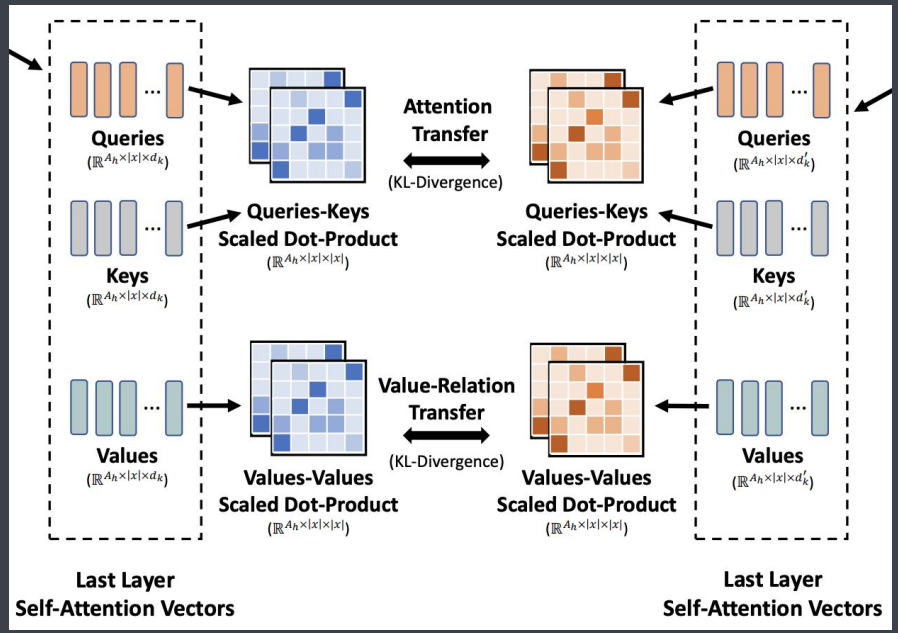
MiniLM

Distillation Method

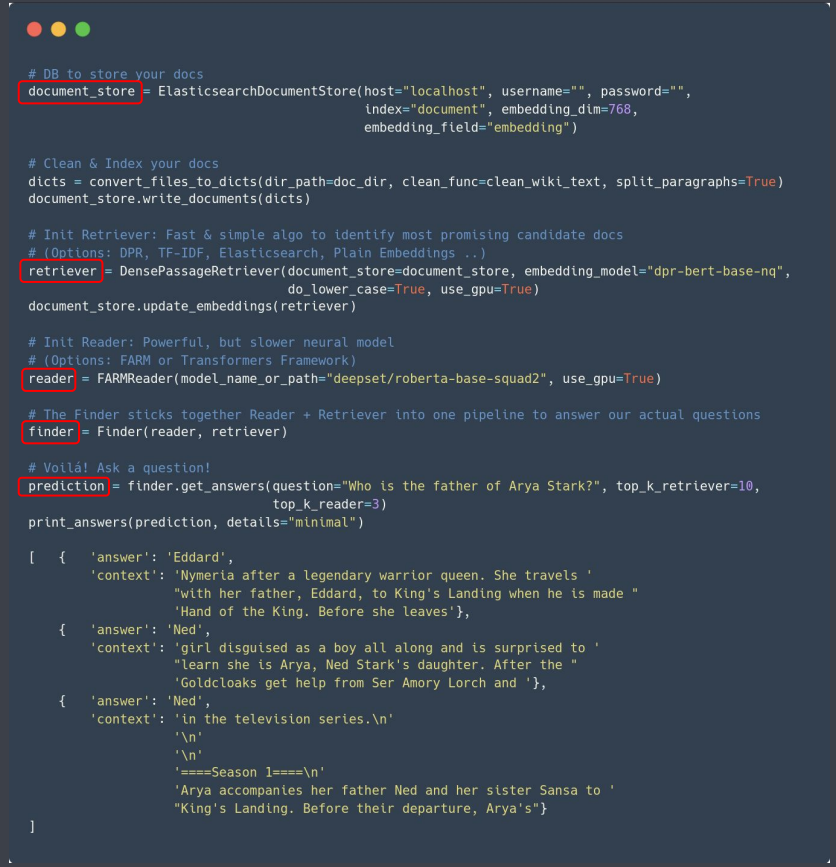
- From UniLM model to 33M parameters (BERT base is 109M)
- 2 types of transfer in last layer

Performance

- Better than BERT base on Glue Benchmark
- 4.9 absolute F1 improvement over BERT base on SQuAD 2
- 1.3x - 2.7x inference speedup on SQuAD 2



Implement it yourself with Haystack



```python
# DB to store your docs
document_store = ElasticsearchDocumentStore(host="localhost", username="", password="",
                                            index="document", embedding_dim=768,
                                            embedding_field="embedding")

# Clean & Index your docs
dicts = convert_files_to_dicts(dir_path=doc_dir, clean_func=clean_wiki_text, split_paragraphs=True)
document_store.write_documents(dicts)

# Init Retriever: Fast & simple algo to identify most promising candidate docs
# (Options: DPR, TF-IDF, Elasticsearch, Plain Embeddings ..)
retriever = DensePassageRetriever(document_store=document_store, embedding_model="dpr-bert-base-nq",
                                  do_lower_case=True, use_gpu=True)
document_store.update_embeddings(retriever)

# Init Reader: Powerful, but slower neural model
# (Options: FARM or Transformers Framework)
reader = FARMReader(model_name_or_path="deepset/roberta-base-squad2", use_gpu=True)

# The Finder sticks together Reader + Retriever into one pipeline to answer our actual questions
finder = Finder(reader, retriever)

# Voilá! Ask a question!
prediction = finder.get_answers(question="Who is the father of Arya Stark?", top_k_retriever=10,
                                top_k_reader=3)
print_answers(prediction, details="minimal")

[   {   'answer': 'Eddard',
        'context': 'Nymeria after a legendary warrior queen. She travels '
                   "with her father, Eddard, to King's Landing when he is made "
                   'Hand of the King. Before she leaves'},
    {   'answer': 'Ned',
        'context': 'girl disguised as a boy all along and is surprised to '
                   "learn she is Arya, Ned Stark's daughter. After the "
                   'Goldcloaks get help from Ser Amory Lorch and '},
    {   'answer': 'Ned',
        'context': 'in the television series.\n'
                   '\n'
                   '\n'
                   '====Season 1====\n'
                   'Arya accompanies her father Ned and her sister Sansa to '
                   "King's Landing. Before their departure, Arya's"}
]
```

What is it?

⟶ Open-Source Framework

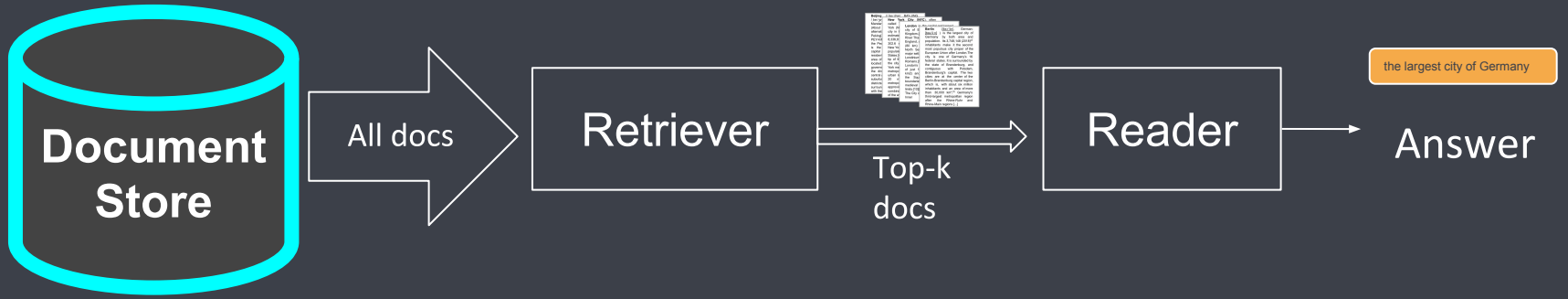
⟶ Modular & easy-to-extend

⟶ Variety of Readers, Retrievers, DocStores

⟶ REST API & Docker for deployment

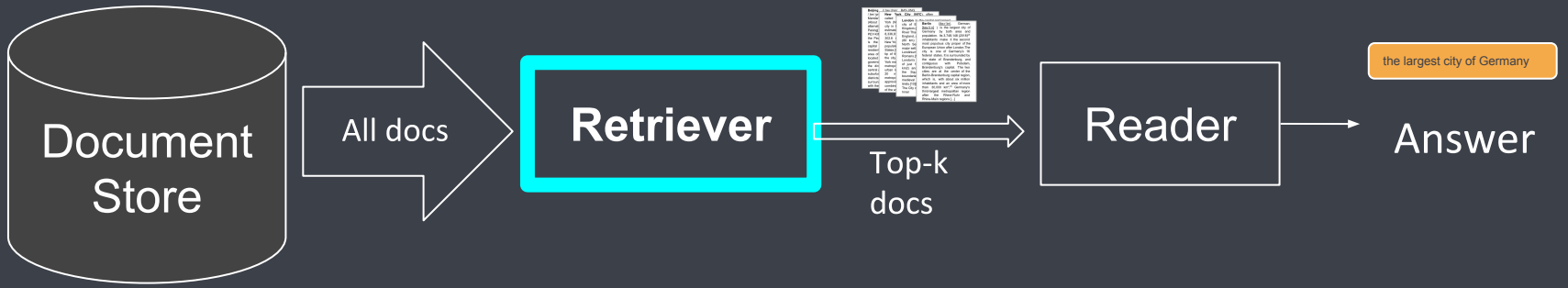
⟶ github.com/deepset-ai/haystack/

What's Available in Haystack



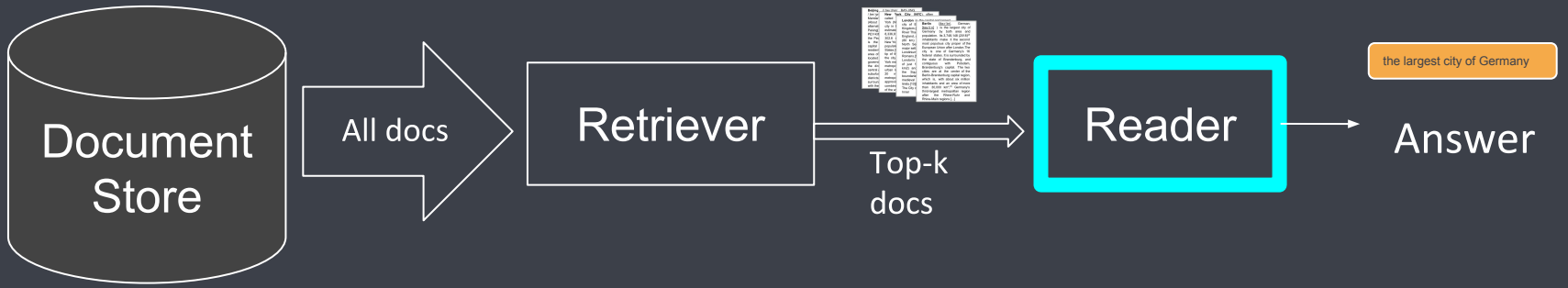
- Elasticsearch
- SQL
- In-memory
- FAISS

What's Available in Haystack



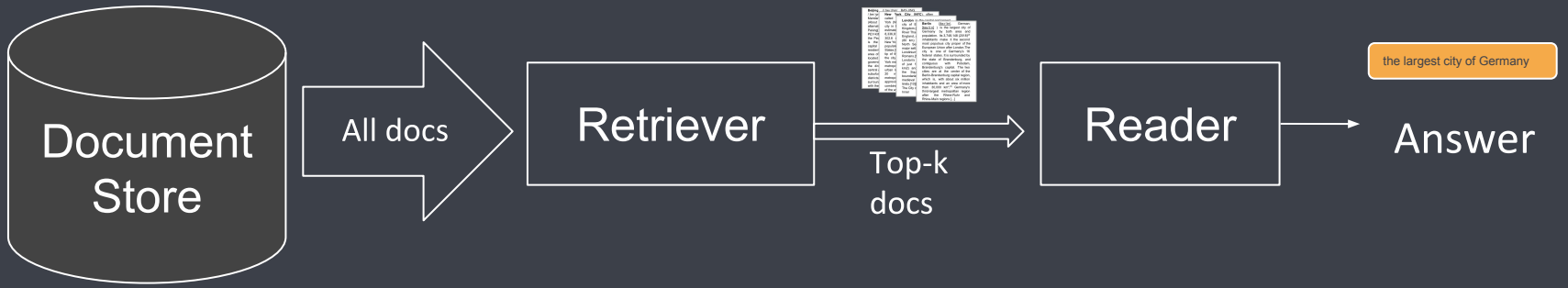
- TF-IDF
- BM25
- Dense Passage Retrieval

What's Available in Haystack



- BERT base
- RoBERTa base
- MiniLM
- Integration with Transformers Model Hub!

What's Available in Haystack



Extras

- Labelling Tool
- Docker Images
- API

01 Language Models

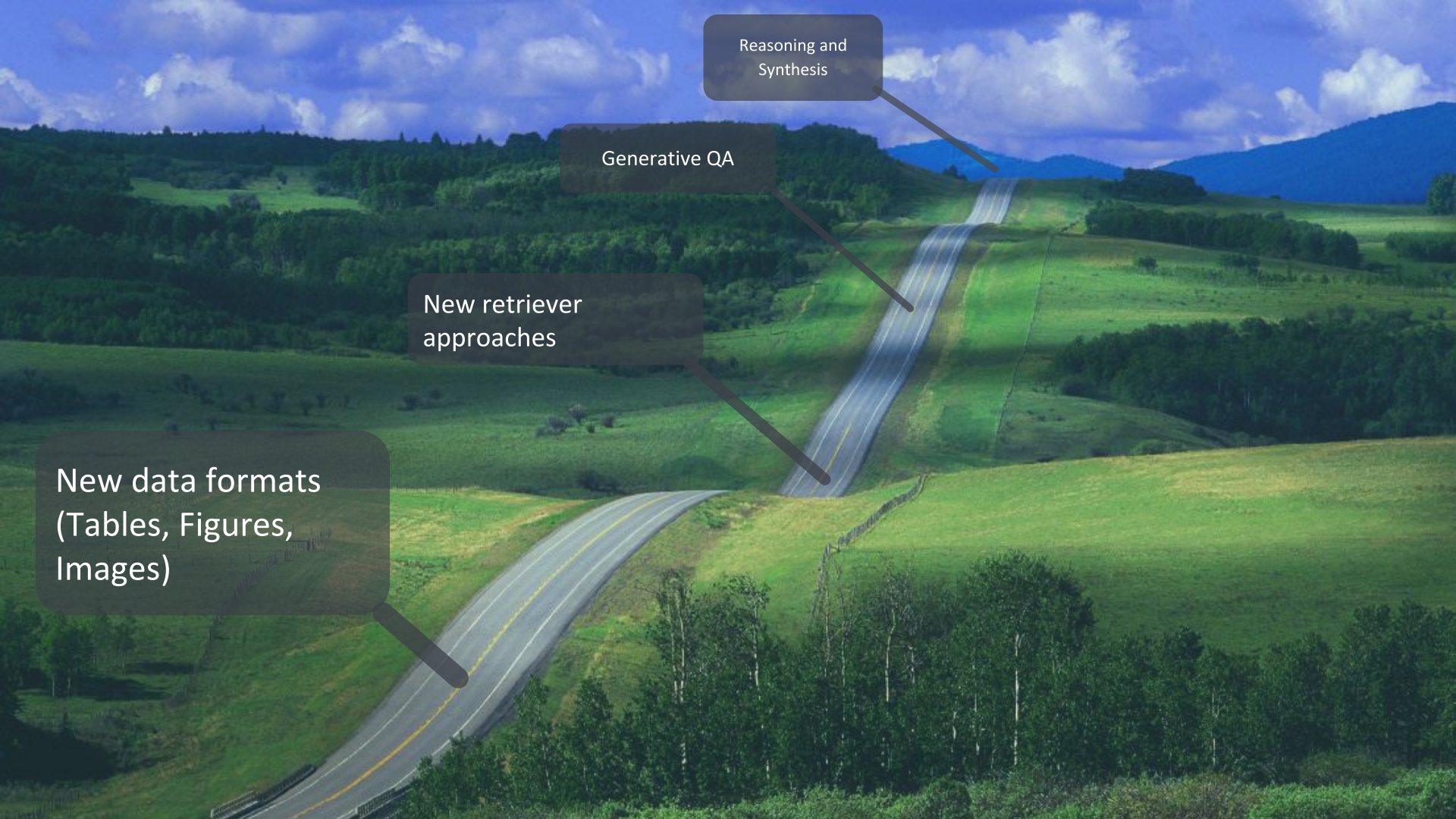
02 Question Answering

03 Scaling: Open-Domain QA

04 Summary

Summary

1. Language Models form the core of most modern NLP Systems
2. QA systems have recently become a lot more powerful
3. Scaling QA systems to an open-domain setting provides a powerful solution to enterprise level information needs
4. You can start building your own end-to-end open-domain QA system with Haystack


Reasoning and Synthesis
Generative QA
New retriever approaches
New data formats (Tables, Figures, Images)

github.com/deepset-ai/**haystack**

@deepset_ai @BrandenChan3

deepset Branden Chan

Join us! We are hiring ...

→ Senior ML Engineer

→ Frontend Developer

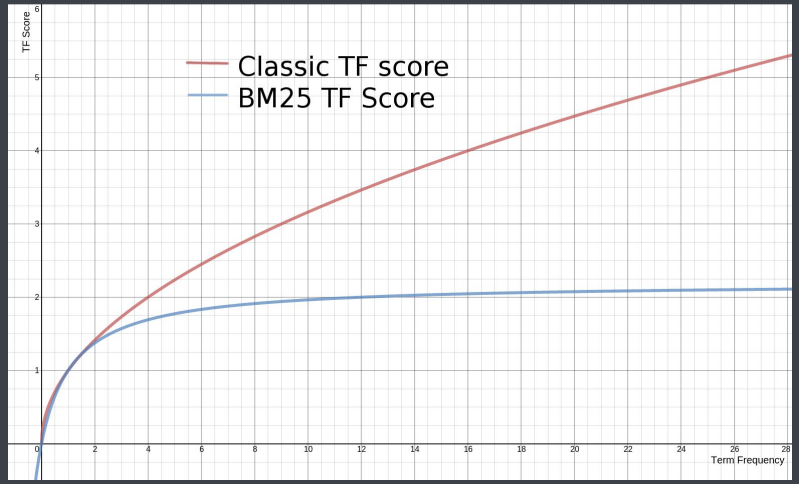
→ Backend Developer

→ DevOps / Solution Architect

BM25

```
BM25-Score = IDF * ((k + 1) * TF) / (k * (1.0 - b + b * (doc_length/avg_doc_length)) + TF)

TF = Term-frequency
IDF = Inverse-document-frequency
k = Saturation of TF (default: 1.2, score saturates after k+1 terms)
b = Impact of doc_length (the higher the better the score for short docs containing the same
TF)
```



Recommended blog article:
https://opensourceconnections.com/blog/2015/10/16/bm25-the-next-generation-of-lucene-relevation/