Building a Question Answering System with Haystack and Elasticsearch

Elvira Siegel,

26.05.2021

What is Haystack?

- Haystack is an open-source framework for building search systems that work intelligently over large document collections.
- Haystack is designed to be the bridge between research and industry.

Haystack

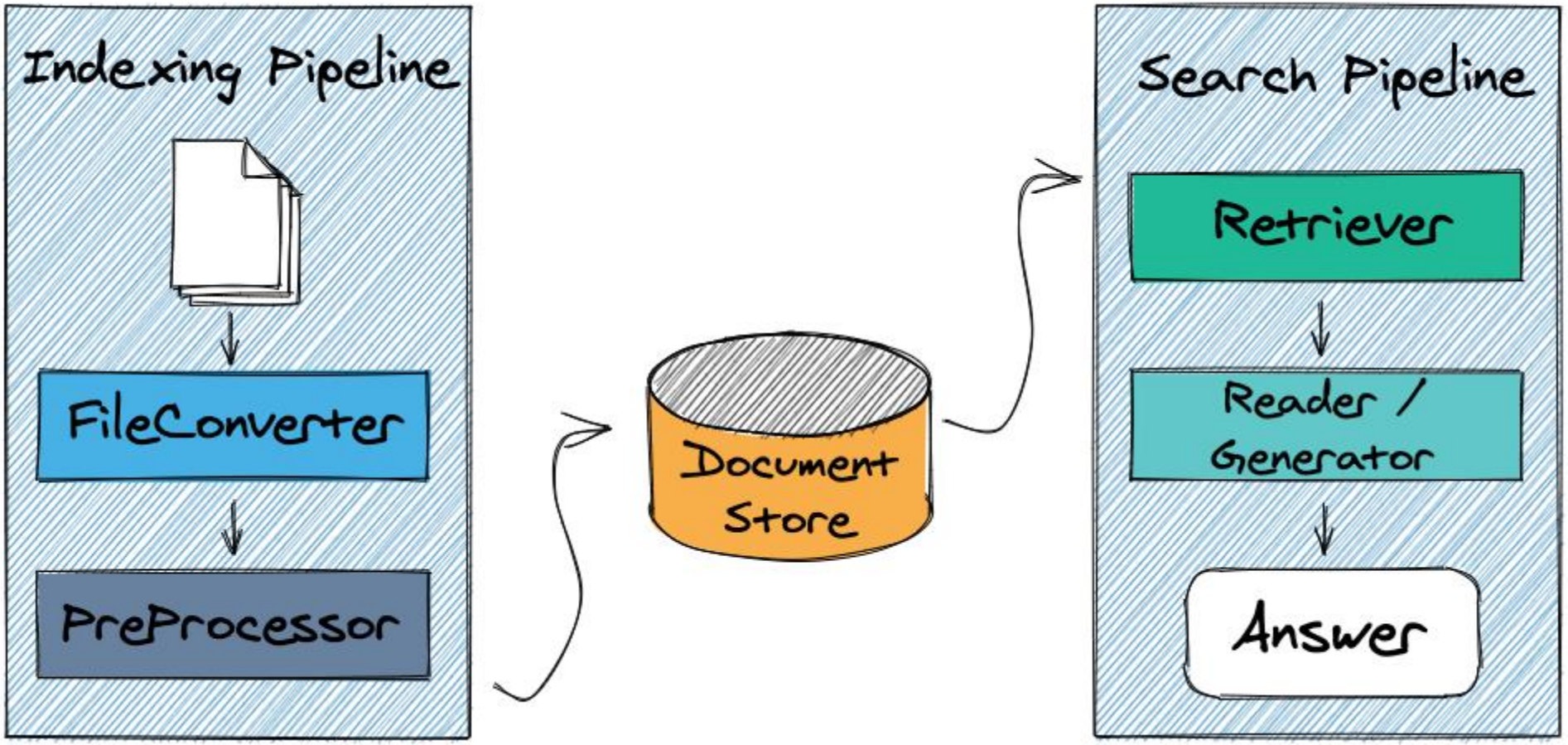
Building a natural language interface for your data

What is Haystack?

It offers...

- *Latest NLP models*: all transformer based models (BERT, RoBERTa, MiniLM, DPR); switch when new ones get published
- *Flexible databases*: load data and query from a range of databases such as Elasticsearch, Milvus, FAISS and SQL
- *Scalability*: Production-ready deployments that scale to millions of documents
- *End-to-End*: all tooling needed to implement,

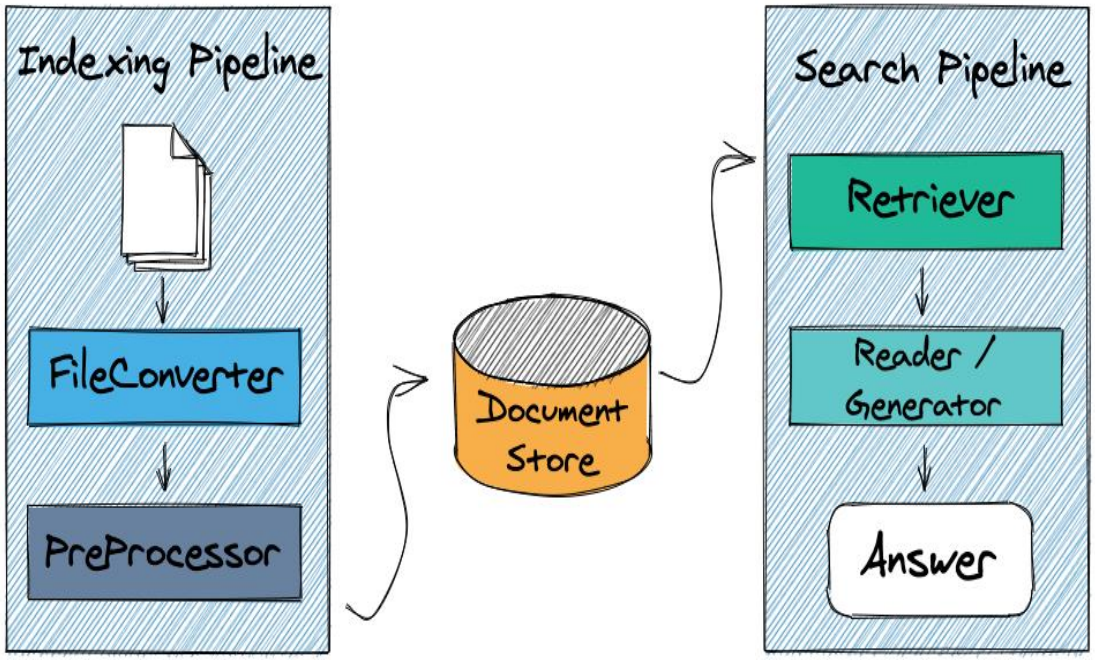
Haystack Structure



Haystack Structure: ***DocumentStore***

DocumentStore: a Database storing the documents for the search.

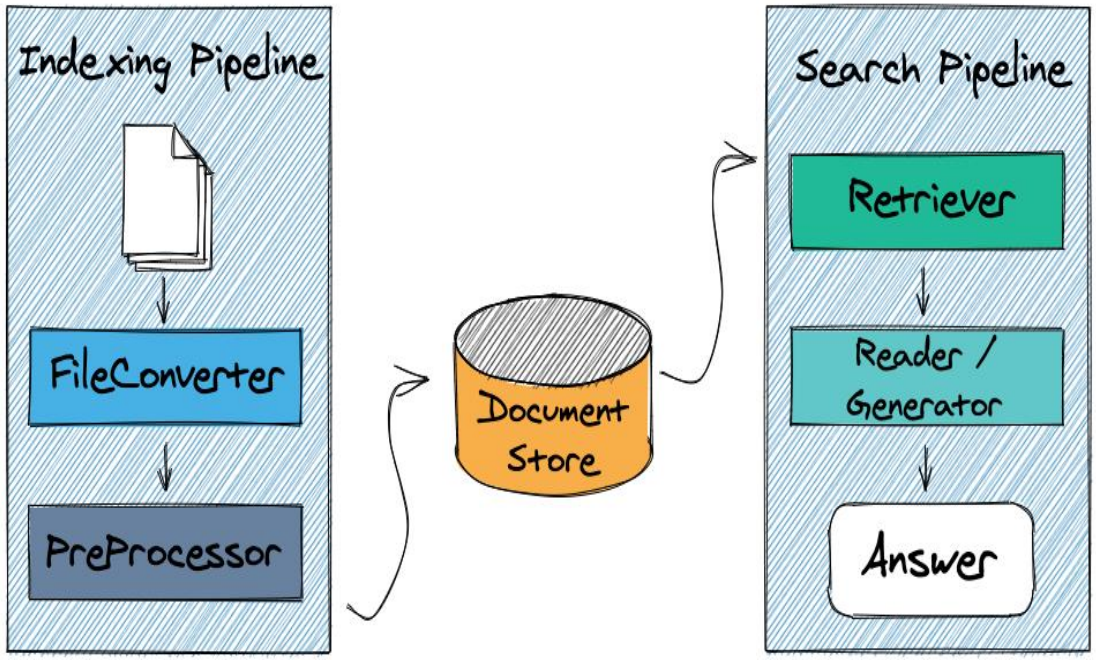
- *Haystack finds answers to queries within the documents stored in a DocumentStore*
- *Elasticsearch or FAISS are given. Also more light-weight options for fast prototyping as SQL or In-Memory*
- *Used in the Project: ElasticsearchDocumentStore*



Haystack Structure: ***Retriever***

Fast algorithms that identify candidate documents for a given query from a large collection of documents.

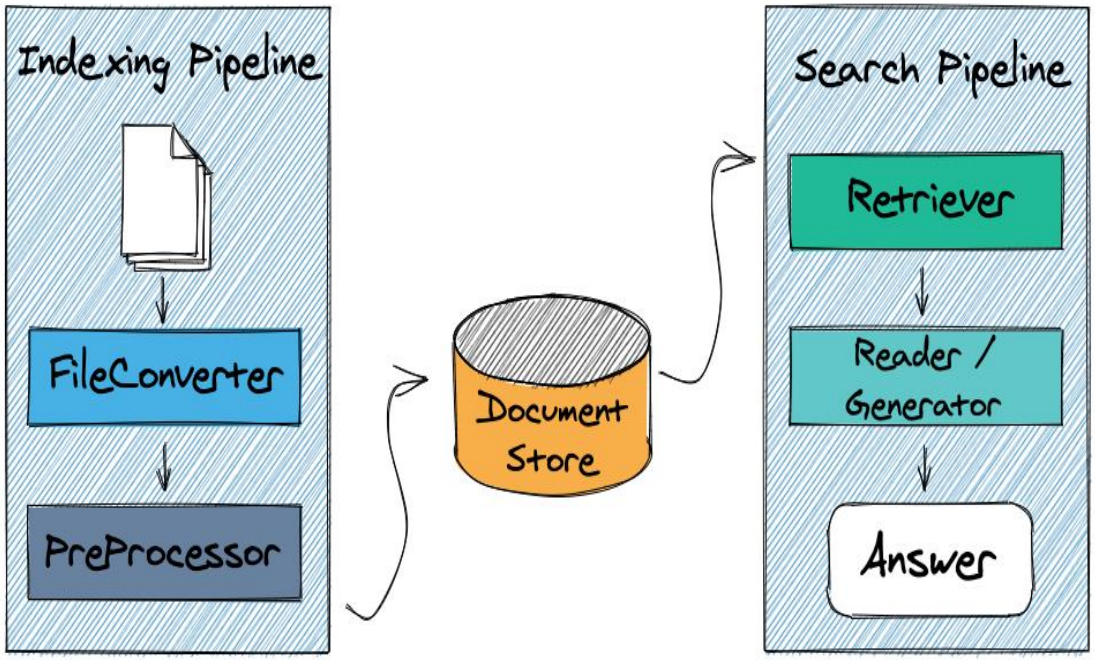
- *Retrievers narrow down the search space and are therefore crucial for scalable QA.*
- *Haystack supports **sparse** methods (TF-IDF, BM25, custom Elasticsearch queries) and state of the art **dense** methods (sentence-transformers and Dense Passage Retrieval)*
- *Retrievers used in the Project: TfidfRetriever and*



Haystack Structure: ***Reader***

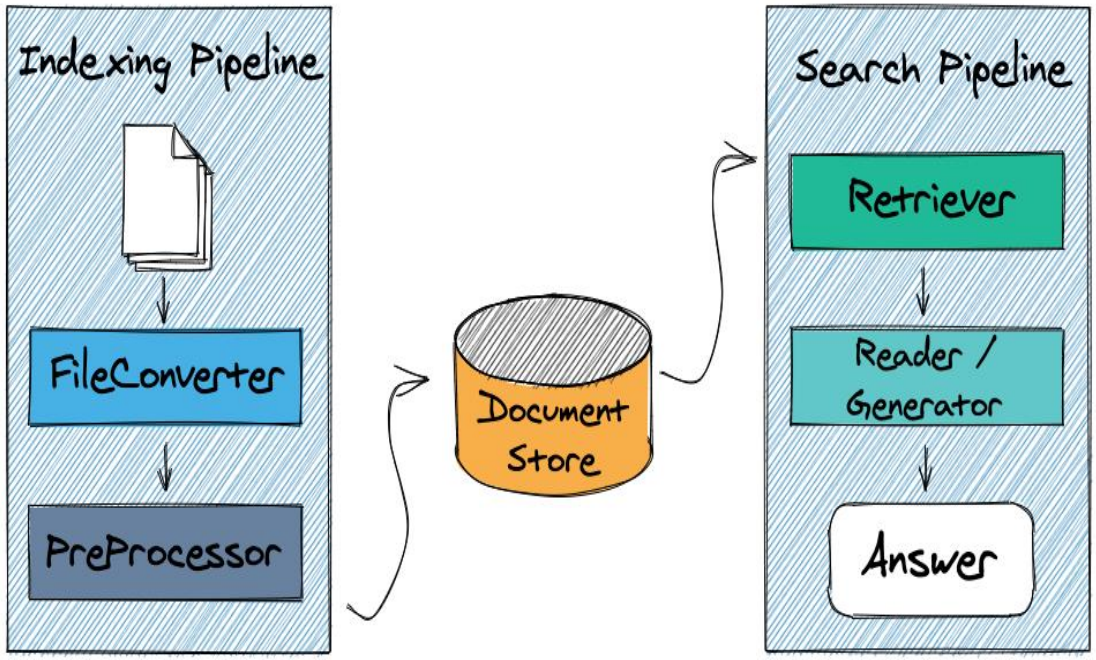
Neural network (BERT, RoBERTa, etc) that reads through texts in detail to find an answer.

- *The Reader takes multiple passages of text as input and returns top-n answers.*
- *Models are trained via FARM or Transformers on SQuAD like tasks. Load a pretrained model from Hugging Face's model hub or fine-tune it on domain data.*
- *Reader used in the Project: FARM - bert-base-multilingual-cased*



Haystack Structure: ***Pipeline***

- *stick together building blocks to a search pipeline.*
- *Pipelines are Directed Acyclic Graphs (DAGs).*
- *Pipeline used in the Project: ExtractiveQAPipeline that combines a retriever and a reader to answer questions.*



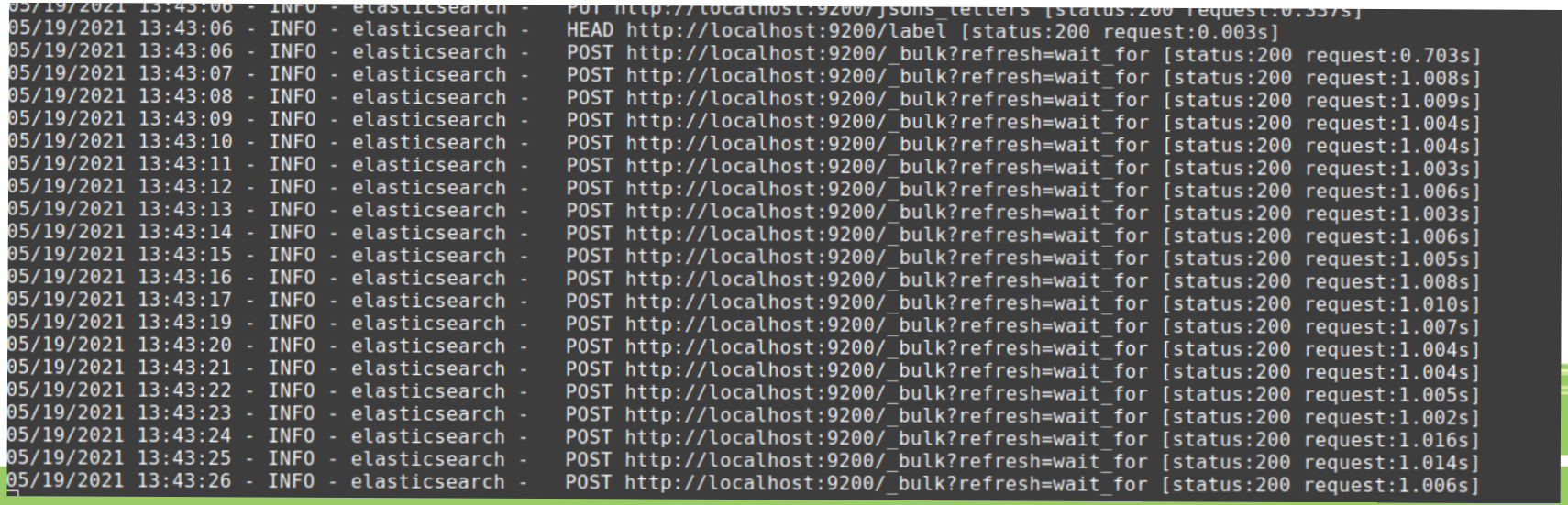
Work with the Letter Collection

- Index the letter collection with ES with the help of Haystack:

- *document_store = **ElasticsearchDocumentStore**(host="localhost", username="", password="", index="jsons_letters")*

- In case, the index does not exist (is empty), fill one with values:

- *document_store.**write_documents**([json_file]) # else use the existing index (Hint: use --to_index)*



TfidfRetriever

- Give the document_store object to:

- *retriever =* ***TfidfRetriever****(document_store=document_store)*

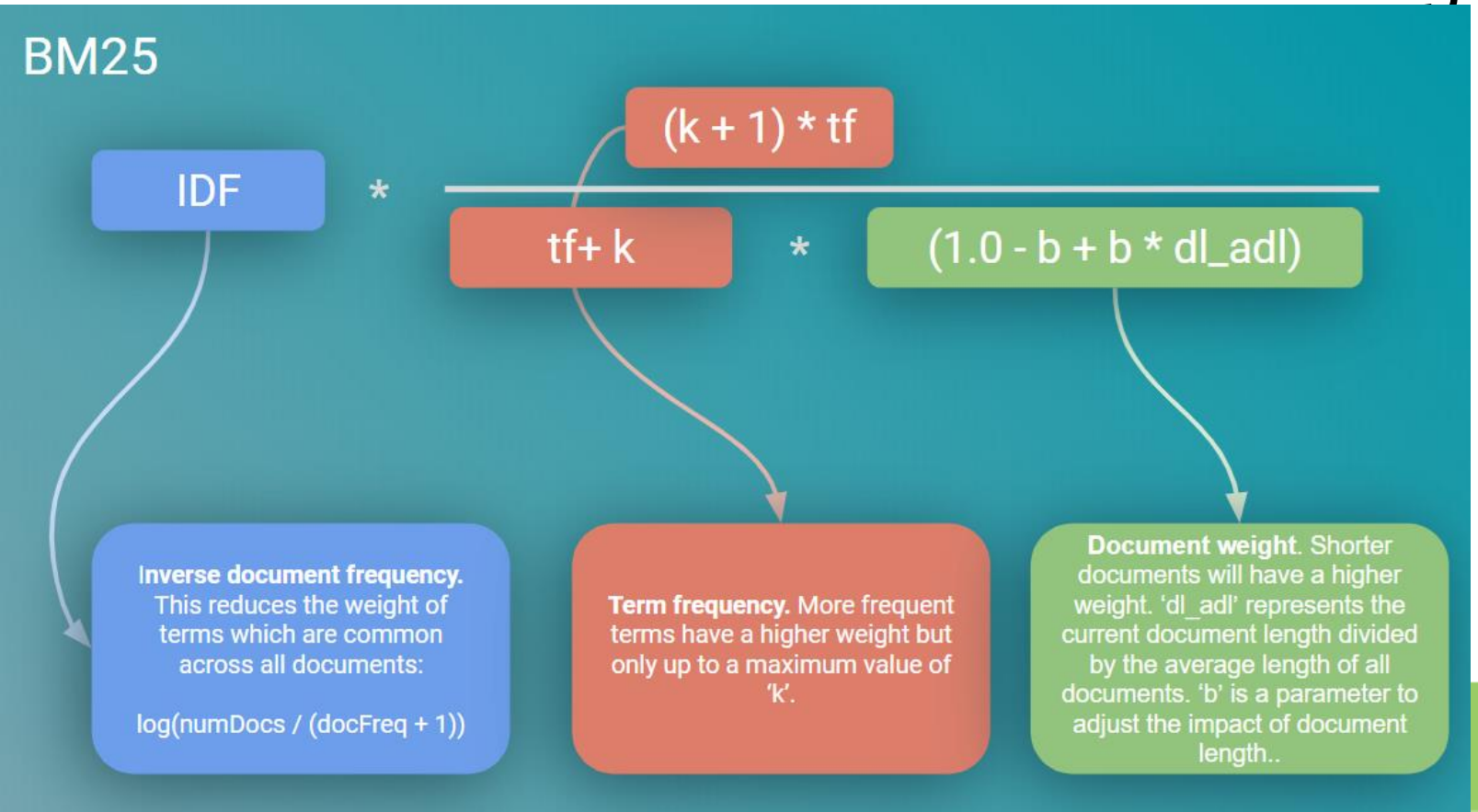
- ***term frequency–inverse document frequency****: reflects how important a word is to a document in a corpus*

$$w_{i,j} = tf_{i,j} \times \log\left(\frac{N}{df_i}\right)$$

$tf_{i,j}$ = number of occurrences of i in j
df_i = number of documents containing i
N = total number of documents

ElasticsearchRetriever

- *retriever = **ElasticsearchRetriever**(document_store=document_store)*
- Here: Elasticsearch's default BM25 algorithm



FARM FARMReader

- FARM is used for Transfer Learning; it is built upon transformers.
- *reader = **FARMReader**(model_name_or_path="bert-base-multilingual-cased", use_gpu=True)*

- Here: *multilingual bert* (mBERT) is used (s. huggingface)

ExtractiveQAPipeline

from haystack.pipeline import ExtractiveQAPipeline

*pipe = **ExtractiveQAPipeline**(reader, retriever)*

Pipeline run

*pipe.run(query=question, **top_k_retriever**=5, top_k_reader=5)*

- Note:
- You can configure how many candidates the reader and retriever shall return: the higher **top_k_retriever**, the better (but also the slower) the answers.
- *run time on average*: between 10 and 20 sec for *top_k_retriever=5*

The Goal

When a question is asked by the user:

The system finds answers within the collection of letters

- What configuration is the best for searching the right answer?

Results (with ***TfidfRetriever*** *and* ***bert-base-multilingual-cased***)

- question: *"Who offered Virginia Woolf a black kitten?"*
- correct answer: *“I have been offered it by Miss Power”*
- *haystack answer*: '… disposition (like mine!) I have been offered it by **Miss Power**, but my room apparently would not suit it, as it … '

Results (with ***TfidfRetriever*** *and* ***bert-base-multilingual-cased****)*

question: *"At what school did James Joyce teach?"*

- correct answer: *"I am an English teacher here in a **Berlitz School**."*

- *haystack answer*: *“It seems we can’t spend Xmas together. All good wishes from us all. Thanks for yours”*

Results *(with ElasticsearchRetriever – **BM25**; and **bert-base-multilingual-cased**)*

question: *"At what school did James Joyce teach?"*

- correct answer: *"I am an English teacher here in a **Berlitz School**."*

- *haystack answer*: (3d result) *'answer': ' …You have asked me to tell you what I am doing and what my prospects are. I am an English teacher here in a **Berlitz School** … '*

Results *(with ElasticsearchRetriever – **BM25**; and **bert-base-multilingual-cased**)*

- question: *"Who offered Virginia Woolf a black kitten?"*
- correct answer: *“I have been offered it by Miss Power”*
- *haystack answer*: *“I’ve not seen or heard from Molly for ages. It seems an unreasonable accusation on her part, considering that the gossip came from so many sources.”*

Results *(with ElasticsearchRetriever – **BM25**; and **bert-base-multilingual-cased**)*

- question: *"Whom met James Joyce at the Liberal Club on 25 January 1903?"*
- correct answer: *"I met **Archer** at the Liberal Club but our talk though it lasted a long time was not very business-like. I also met **Lady Gregory** and had just time to see Mr O'Connell before I caught my train."*

- *haystack answer*: "*I met **Archer** at the Liberal*

Results (with ***TfidfRetriever*** *and* ***bert-base-uncased****)*

question: *"At what school did James Joyce teach?"*

- correct answer: *"I am an English teacher here in a **Berlitz School**."*

- *haystack answer*: 'Zug? In fact, ***any school*** whatever which can pay a small monthly [salary].'

Results (with **ElasticsearchRetriever** *and* ***bert-base-uncased****)*

question: *"At what school did James Joyce teach?"*

- correct answer: *"I am an English teacher here in a **Berlitz School**."*

- *haystack answer*: 'I am an English teacher here in a ***Berlitz School***. I have been here for sixteen months during which time I have achieved the delicate task of living and ...'

Results (with **ElasticsearchRetriever** *and* ***bert-base-german-cased****)*

question: *"Der Kampf ums Dasein heißt nach Freud?"*

- correct answer: “*Der ›Kampf ums Dasein‹ heißt für mich noch ein Kampf ums ›Dableiben‹.”*

- *haystack answer*: 'Ich habe fast *gar nichts arbeiten können, dank meiner ärztlichen Tätigkeit …'*

Results

- Specific language models *(e.g. German BERT, English BERT)* do not necessary make the prediciton better.
- BM25 gives better predictions than TFIDF
- BM25 and mBERT is the recommended configuration to start with (in this project specifically)

The GitHub Project

- GitHub Project (Code): https://github.com/elliesiegel/Haystack_ElasticSeach

Sources

- Haystack Project:
- https://github.com/deepset-ai/haystack
- Haystack Pipelines:
https://haystack.deepset.ai/docs/latest/pipelinesmd
- HuggingFace Model Hub:
- https://huggingface.co/models
- FARM: https://github.com/deepset-ai/FARM

Thank You!