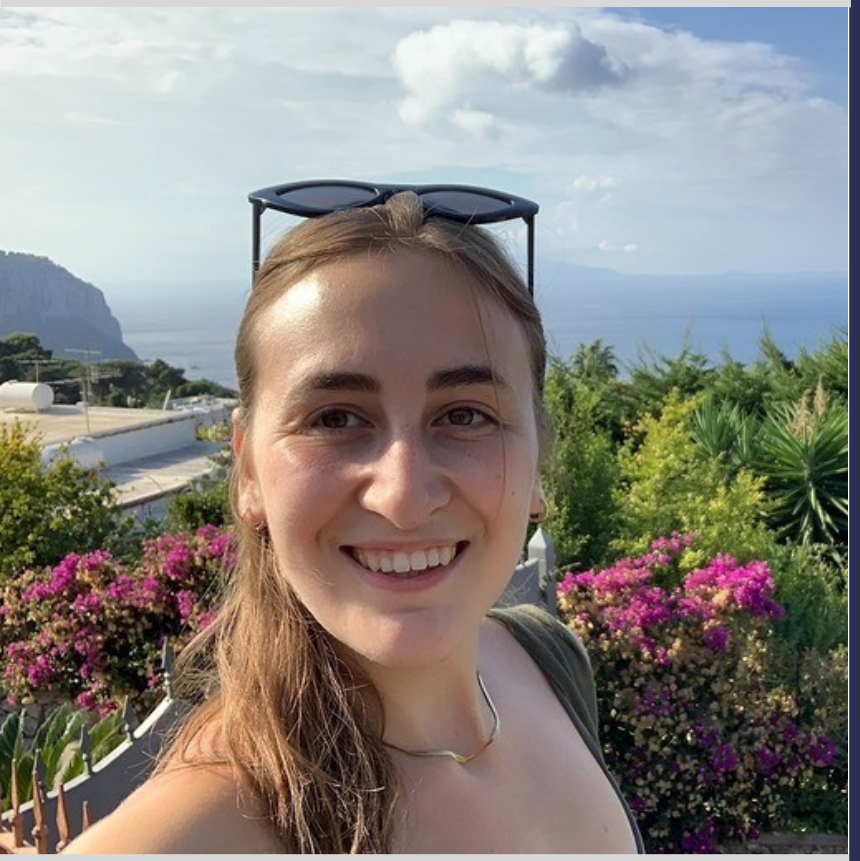
Build a Semantic Search Application in Python

Using Haystack

FOSDEM 2023 Tuana Celik February 4th, 2023



Tuana Celik

Developer Advocate

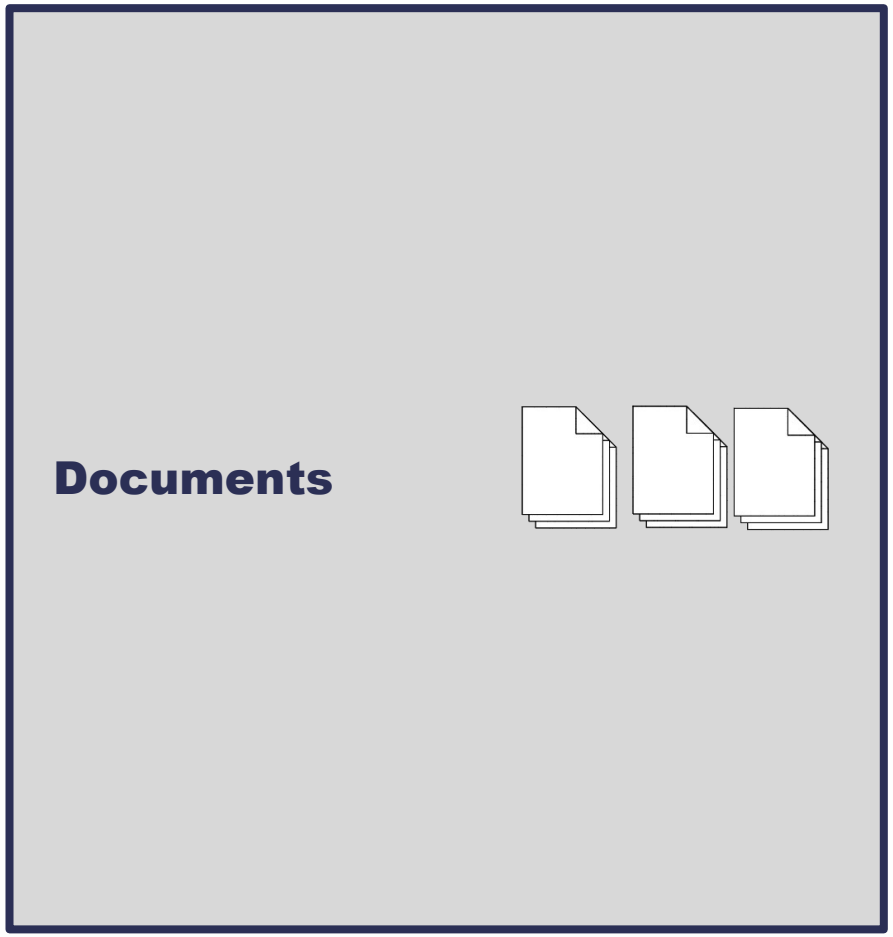
deepset

- MEng Computer Science (Bristol)
- DevRel since 2020
- At deepset since 2022
- Twitter: @tuanacelik
- GitHub: @TuanaCelik

Agenda

1	A brief history: Keyword Search to Semantic search
2	How to implement semantic search
3	Example with Haystack

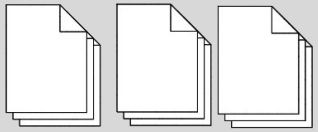
Keyword Search



? Query - “Kardashian sisters”

Keyword Search

Documents



The **Kardashian family**, also referred to as the **Kardashian–Jenner family**, is an American **family** prominent in the fields of entertainment, reality television, fashion design, and business. Founded by Robert **Kardashian** and Kris Jenner, it consists of their children Kourtney, Kim, Khloé, and Rob Kardashian, as well as their grandchildren. After Robert and Kris' divorce in 1991, Kris married Caitlyn Jenner, with whom she had two daughters: Kendall and Kylie Jenner. Notable extended relatives include Kendall and Kylie's half-**siblings**

? Query - “Kardashian sisters”

Keyword matching

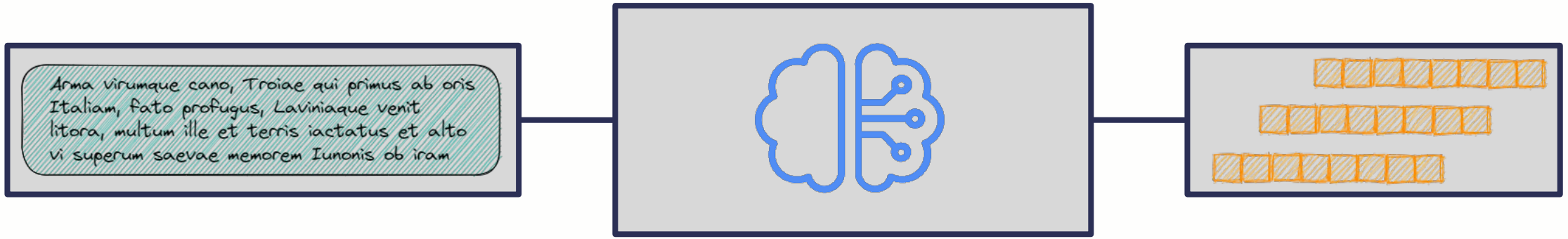
Semantic Search

- What if that's just not enough?
- What if I want to be able to make my query "Who is the richest Kardashian sister?"
- How will it *get* what I'm trying to ask..

Queue: Language Models

the evolution

Vectors, or ‘Embeddings’



Transformer language models are amazing

- Neural networks trained on large amounts of text
- Turns text into vectors
- Can generalize quite well

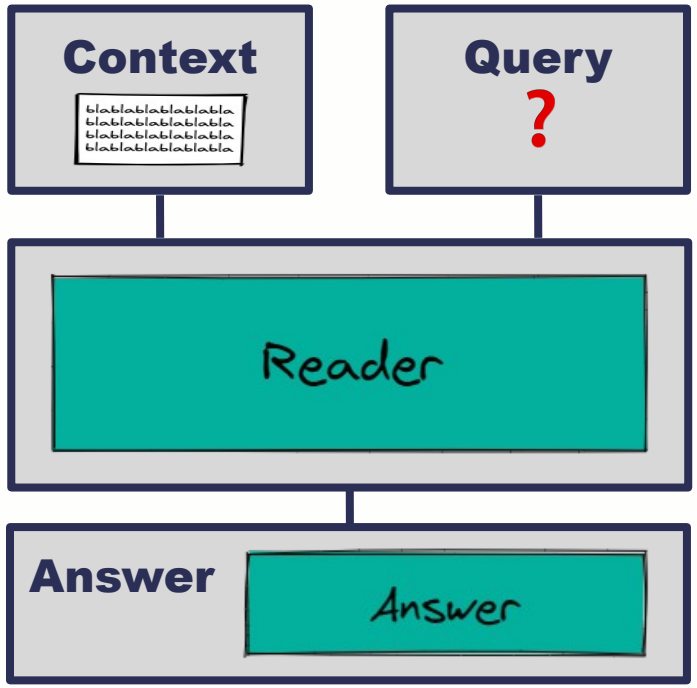
They enable different NLP applications

- QA, Summarization, Retrieval, Translation, Reranking

Transformer based NLP will be a part of every application!

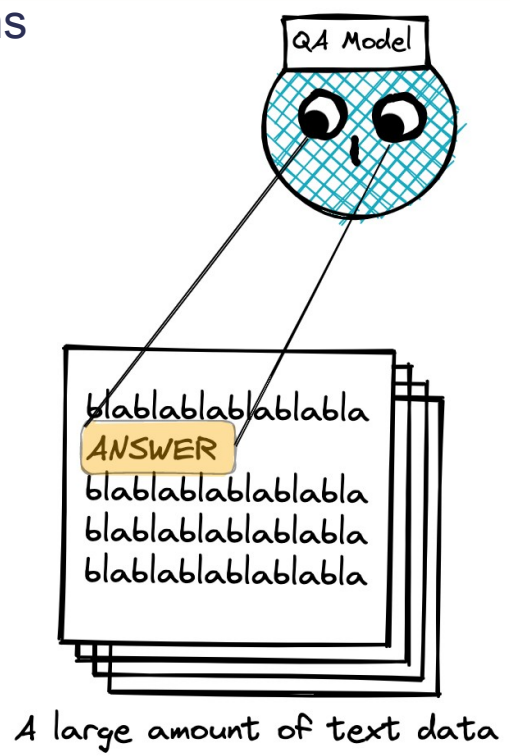
- Already used in Google Search, Translate, Product recommendations…

The Evolution From Extractive to Generative Models

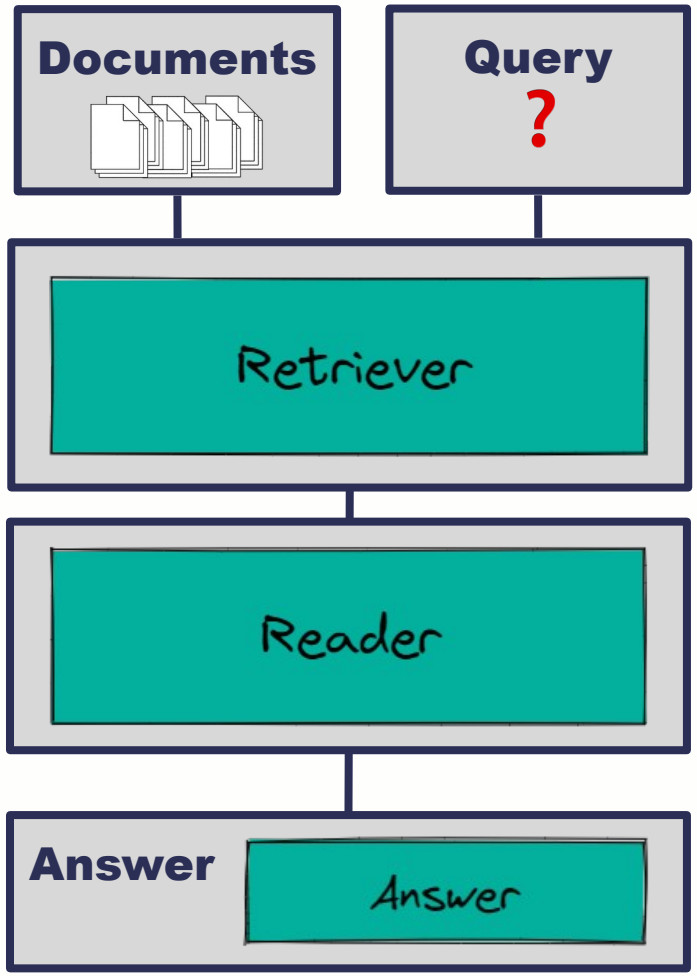


Challenges for the Reader

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



The Evolution From Extractive to Generative Models



Retriever + Reader

- Efficient for large collections of text
- Retriever acts as lightweight filter
- Passes candidates to Reader

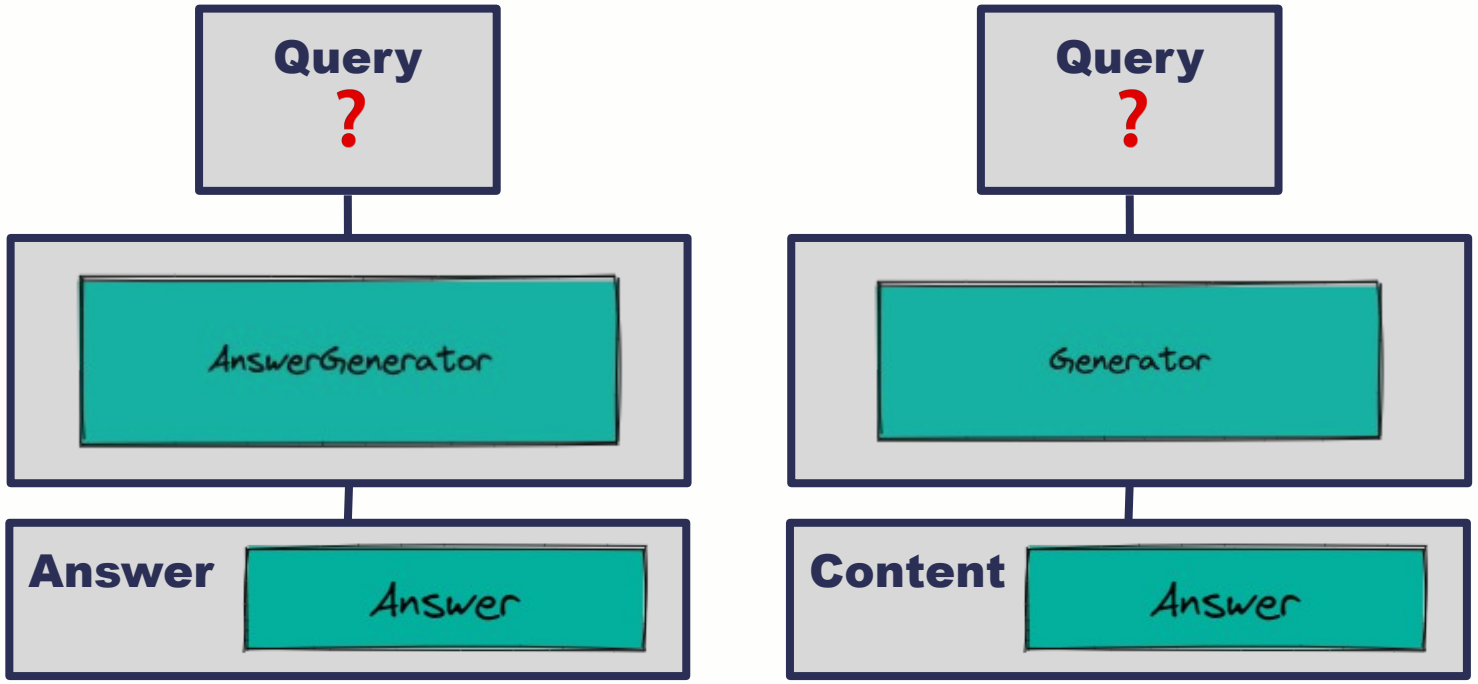
Reader: 3+ hours / query
w/ Retriever: 0.5-2 sec / query

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

The Evolution From Extractive to Generative Models

Generative Models

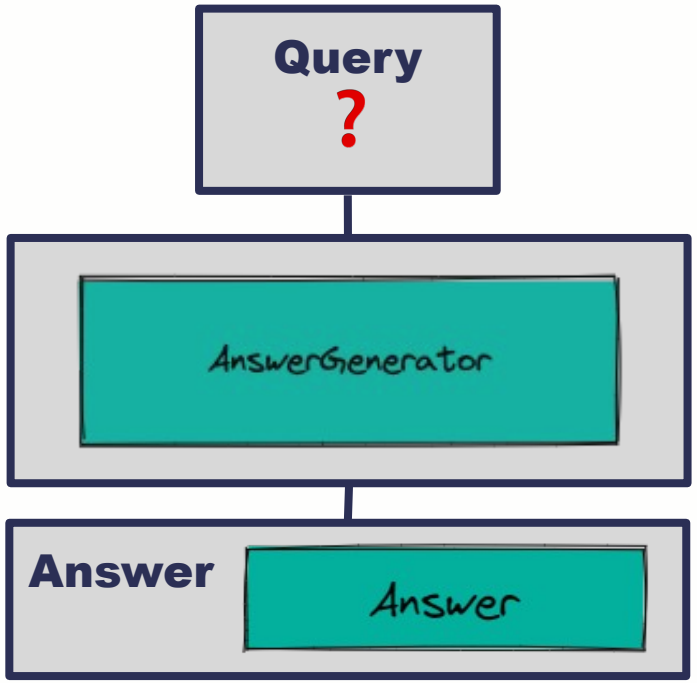
- Don’t need context
- Produce human-like answers

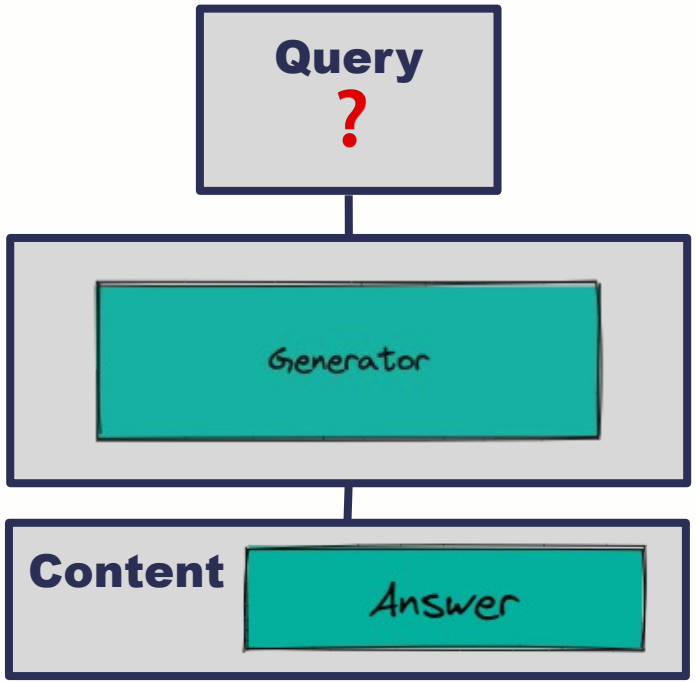


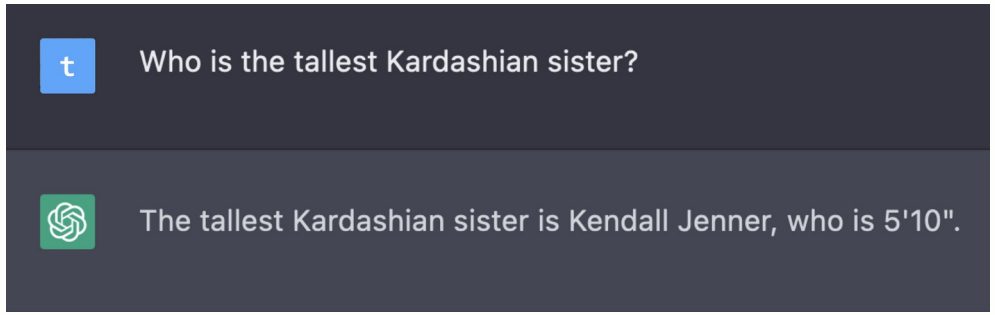
The Evolution From Extractive to Generative Models

Generative Models

- Don’t need context
- Produce human-like answers



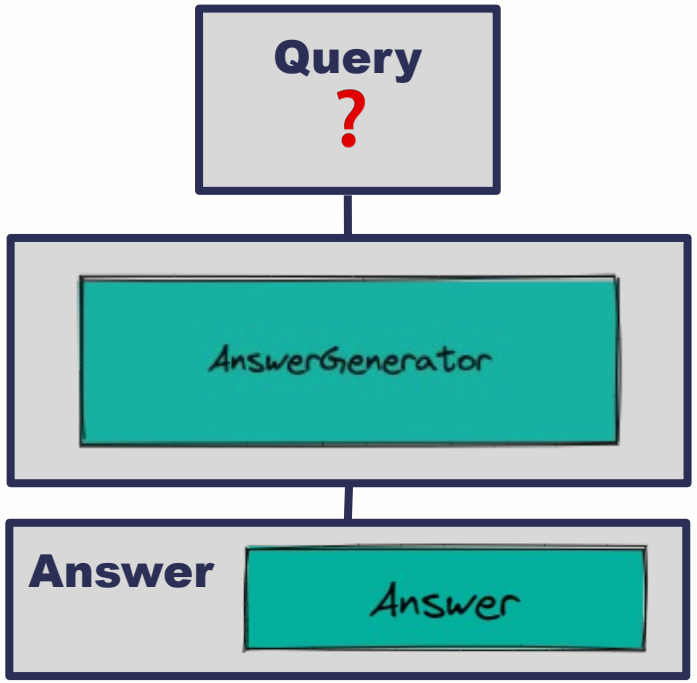


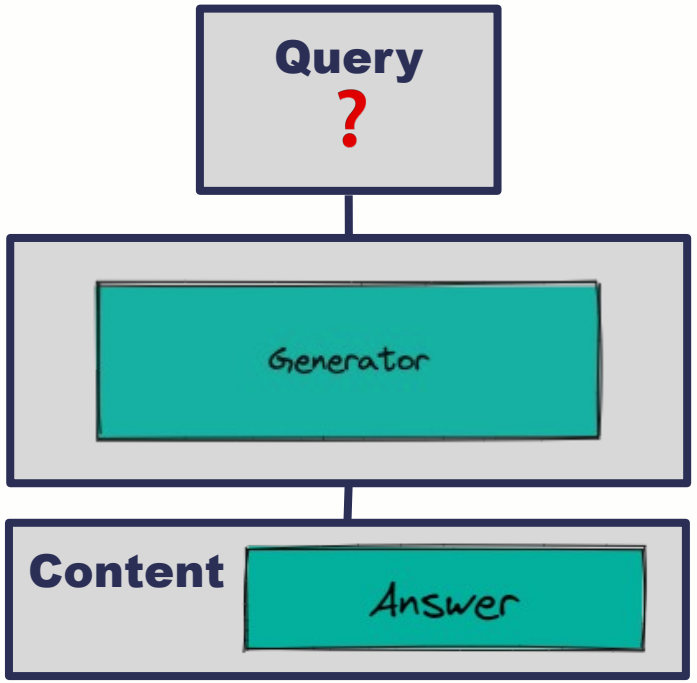


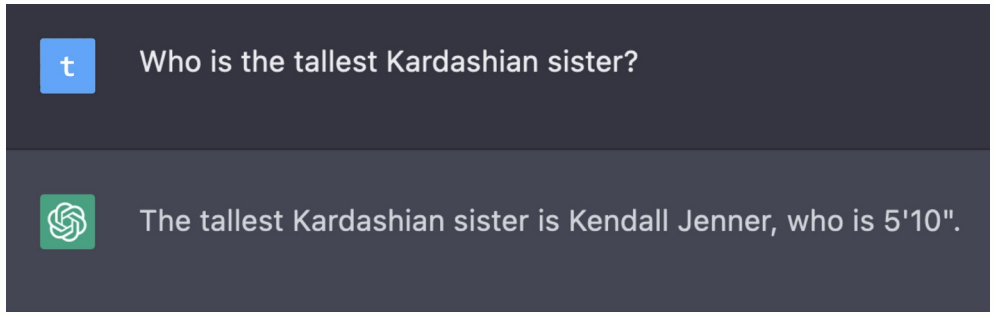
The Evolution From Extractive to Generative Models

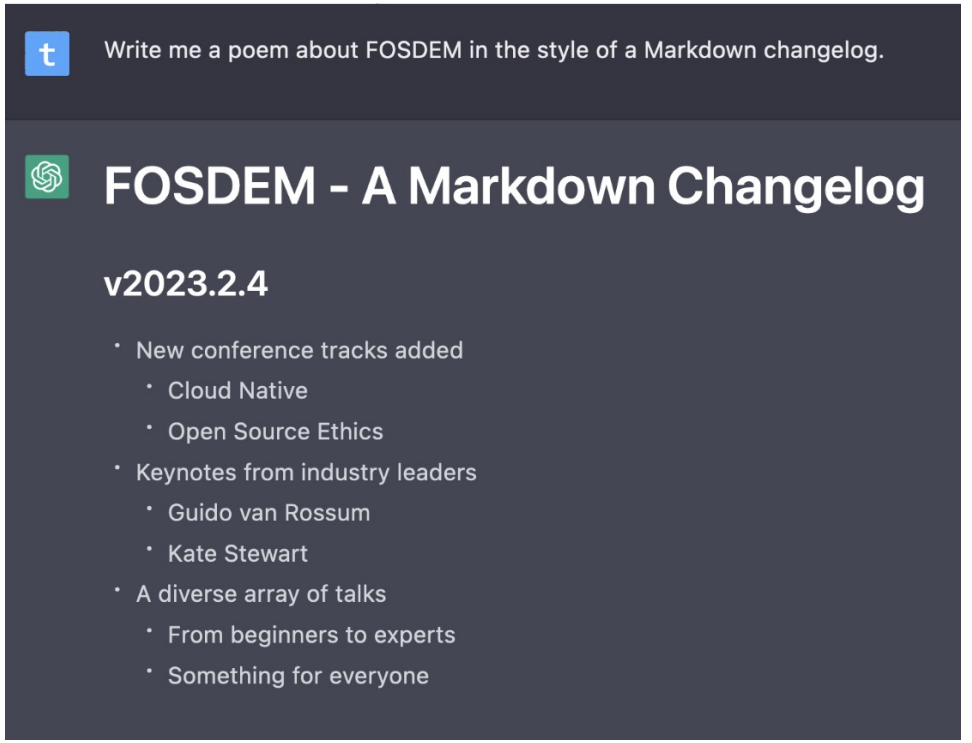
Generative Models

- Don’t need context
- Produce human-like answers

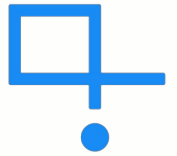








The Evolution From Extractive to Generative Models



These Language Models are readily available

- Companies like OpenAI and Cohere provide increasingly impressive LLMs
- There are many many open source models readily available on Hugging Face
- These models bridge the gap between a result being just a ‘search result’ to an actual human-like answer

Next steps:

- **How do we use these language models for various use cases?**

haystack
by deepset

What is Haystack?

- Fully open source NLP framework built in Python
- The core NLP tasks covered
- Production focused

Integration with HF model hub

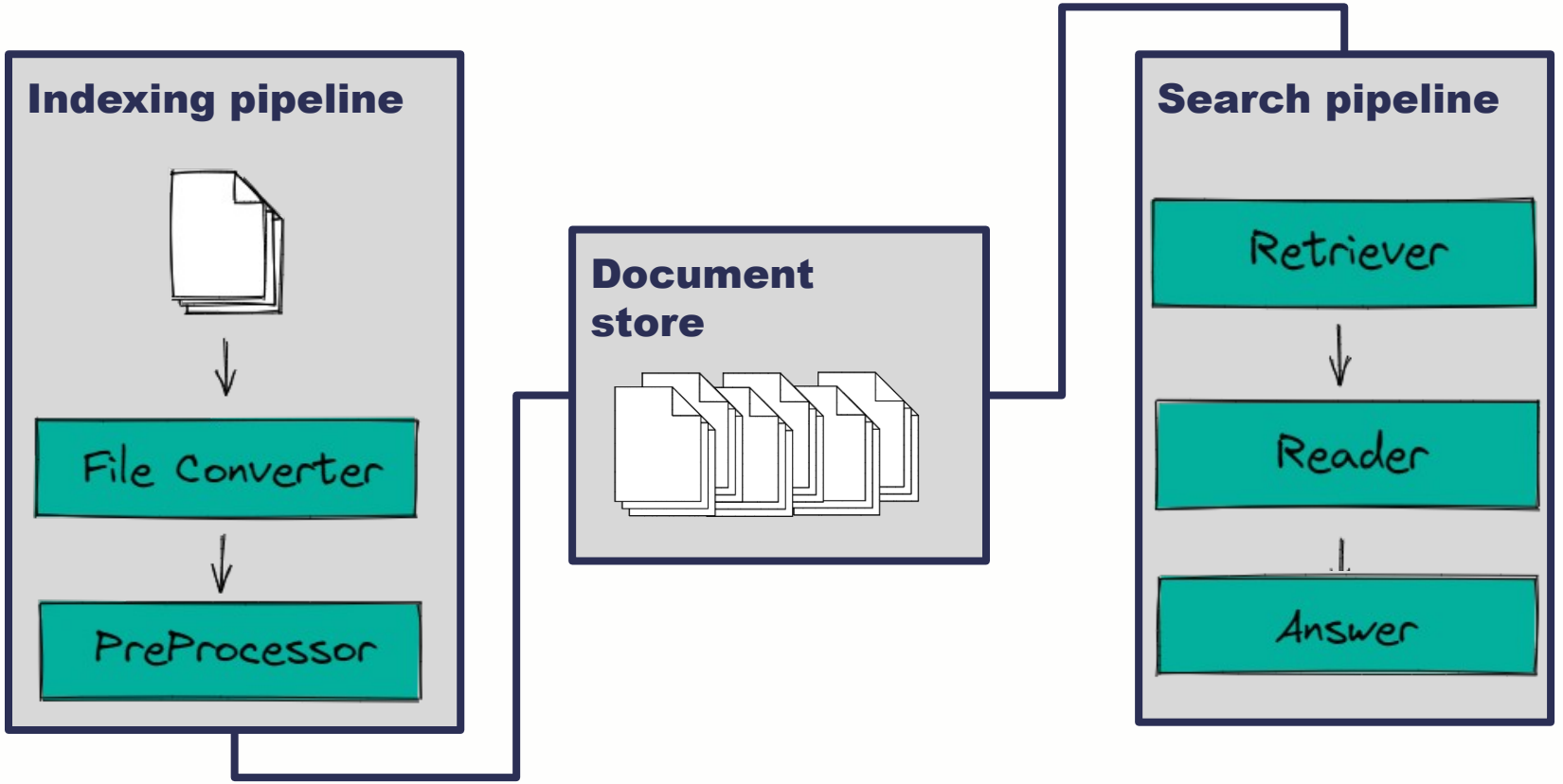
- Easy to use the latest models

Integration with OpenAI and Cohere

- Use your API Key to plug in LLMs in your applications

It's end-to-end, not just models

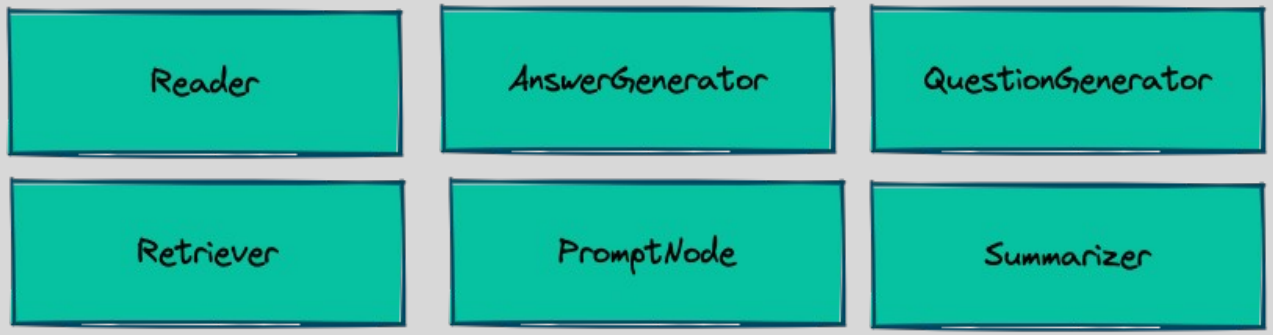
- Connector between components
- Efficient storage options
- QoL features for all users



haystack
by deepset

Features of Haystack

Core NLP Tasks covered



Data Connectors

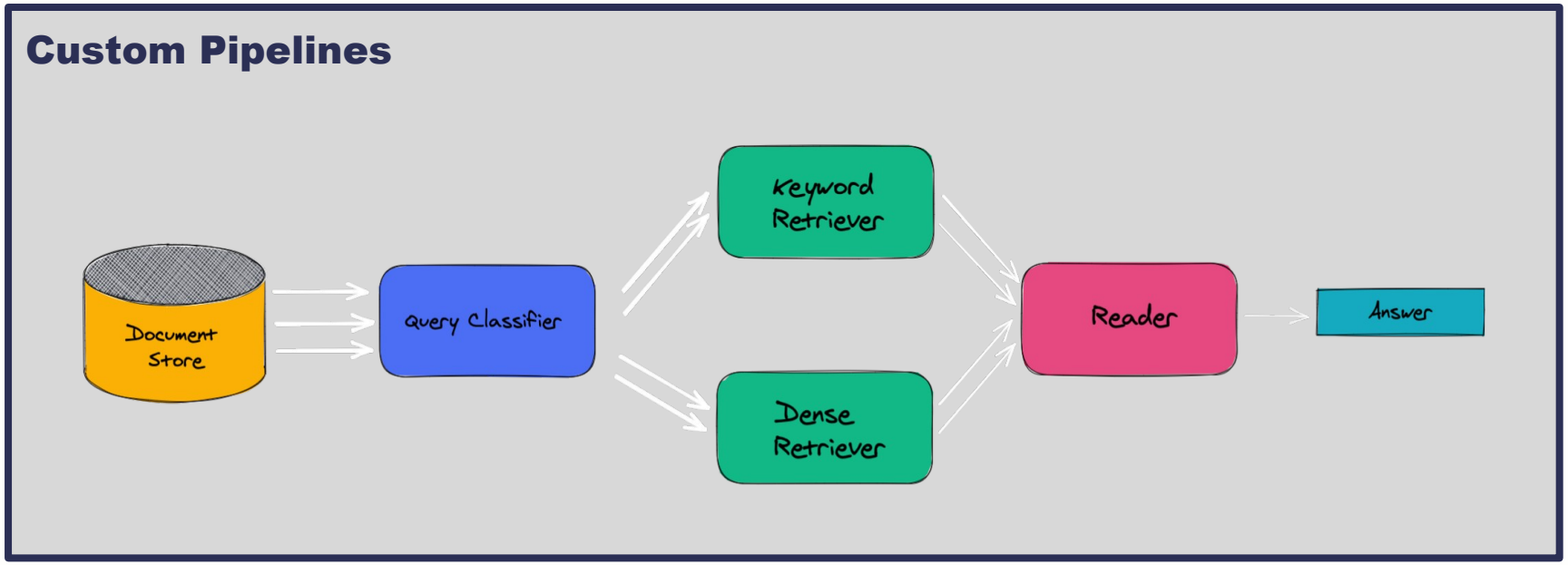
- File converters from txt, pdf, docx, markdown
- Web scraper to turn website into text
- Preprocessor to split long documents, clean text

Document stores

- Elasticsearch, OpenSearch, SQL (standard options)
- FAISS, Pinecone, Milvus, Weaviate (vector optimized)

haystack
by deepset

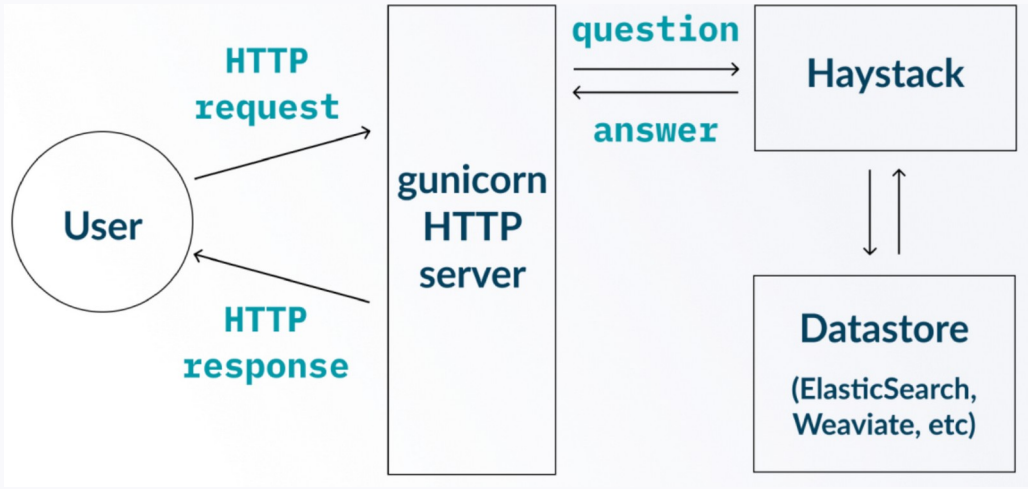
Features of Haystack



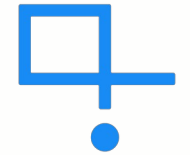
haystack
by deepset

Features of Haystack

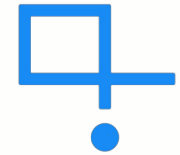
Expose your applications via REST API



```
pip install farm-haystack
```

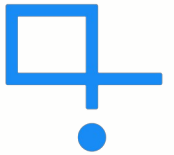


```
pip install farm-haystack
```

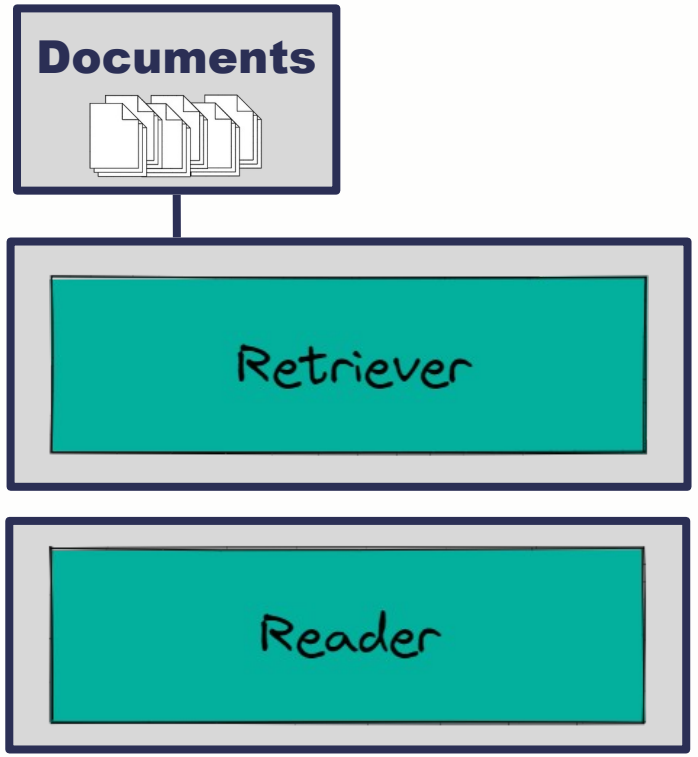


```python
from haystack.nodes.retriever import EmbeddingRetriever
from haystack.nodes.reader import FARMReader

# Component to retrieve documents relevant to a query
retriever = EmbeddingRetriever(document_store=document_store, embedding_model='some_model/name')

# Component to extract answers to a query from documents
reader = FARMReader(model_name_or_path='some_other_model/name')
```

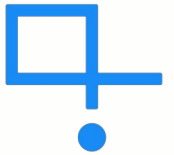
```
pip install farm-haystack
```



```
from haystack.nodes.retriever import EmbeddingRetriever
from haystack.nodes.reader import FARMReader

# Component to retrieve documents relevant to a query
retriever = EmbeddingRetriever(document_store=document_store, embedding_model='some_model/name')

# Component to extract answers to a query from documents
reader = FARMReader(model_name_or_path='some_other_model/name')
```

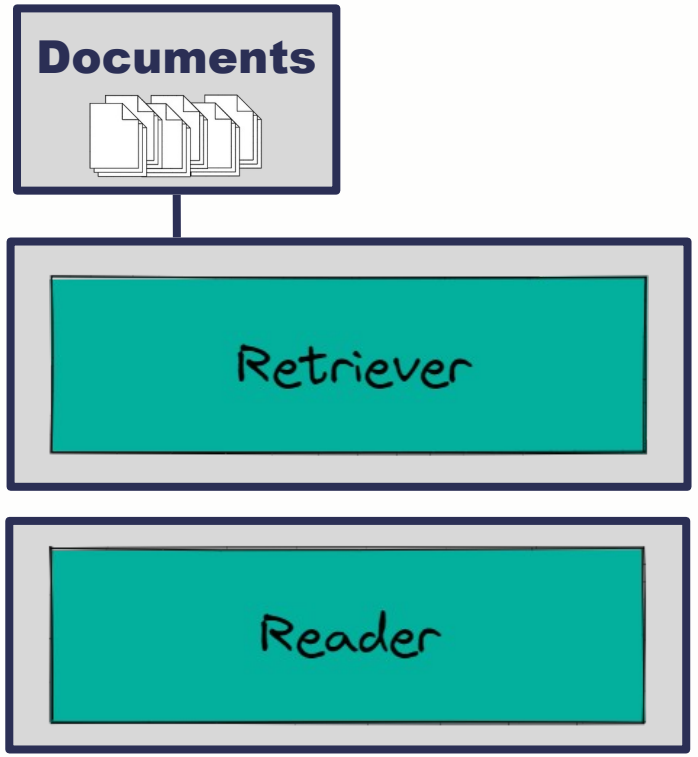


```
pip install farm-haystack
```



```
from haystack.nodes.retriever import EmbeddingRetriever
from haystack.nodes.reader import FARMReader

# Component to retrieve documents relevant to a query
retriever = EmbeddingRetriever(document_store=document_store, embedding_model='some_model/name')

# Component to extract answers to a query from documents
reader = FARMReader(model_name_or_path='some_other_model/name')
```

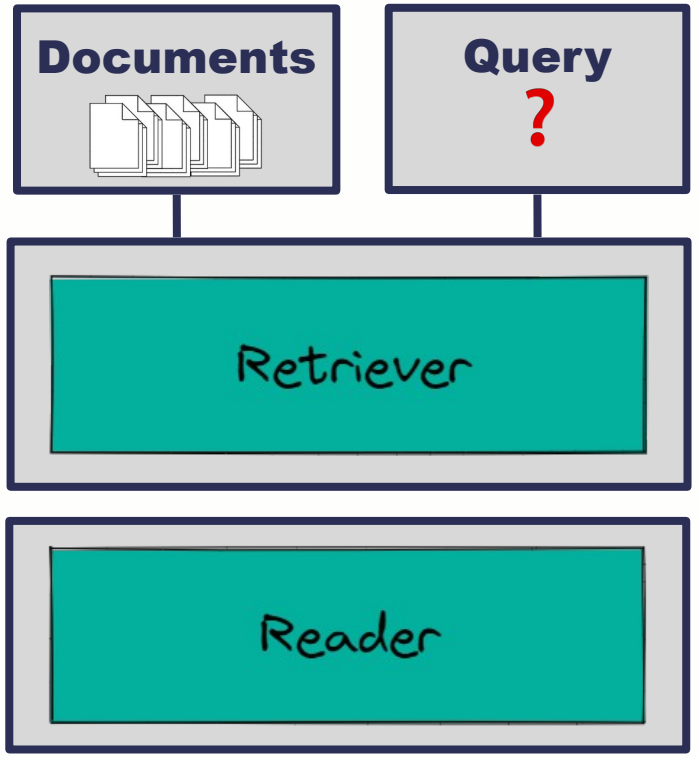
```
from haystack import Pipeline

pipeline = Pipeline()

pipeline.add_node(component=retriever, name="Retriever", inputs=["Query"])
```



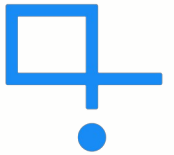
```
pip install farm-haystack
```

```
from haystack.nodes.retriever import EmbeddingRetriever
from haystack.nodes.reader import FARMReader

# Component to retrieve documents relevant to a query
retriever = EmbeddingRetriever(document_store=document_store, embedding_model='some_model/name')

# Component to extract answers to a query from documents
reader = FARMReader(model_name_or_path='some_other_model/name')
```

```
from haystack import Pipeline

pipeline = Pipeline()

pipeline.add_node(component=retriever, name="Retriever", inputs=["Query"])
```

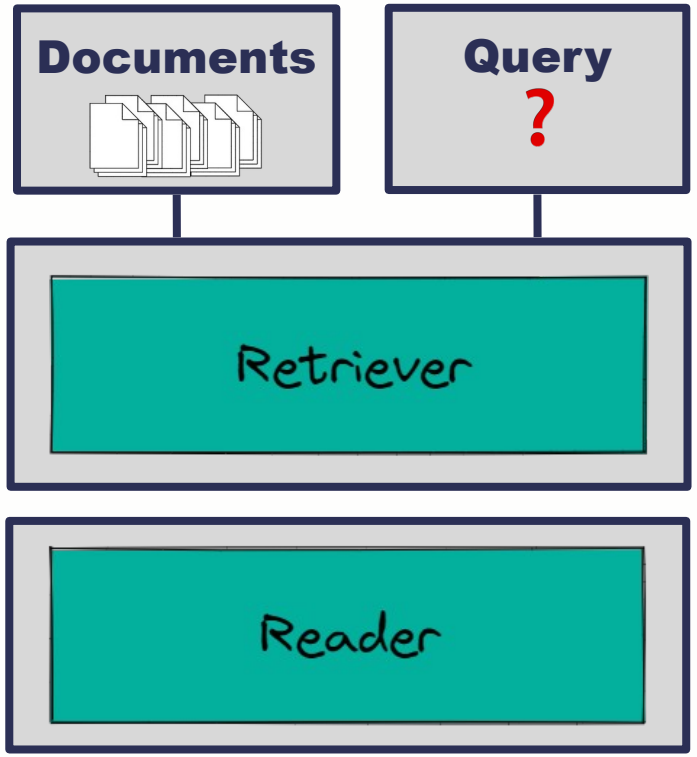


```
pip install farm-haystack
```

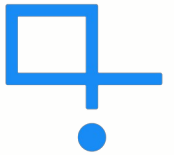


```
from haystack.nodes.retriever import EmbeddingRetriever
from haystack.nodes.reader import FARMReader

# Component to retrieve documents relevant to a query
retriever = EmbeddingRetriever(document_store=document_store, embedding_model='some_model/name')

# Component to extract answers to a query from documents
reader = FARMReader(model_name_or_path='some_other_model/name')
```

```
from haystack import Pipeline

pipeline = Pipeline()

pipeline.add_node(component=retriever, name="Retriever", inputs=["Query"])


pipeline.add_node(component=reader, name="Reader", inputs=["Retriever"])
```

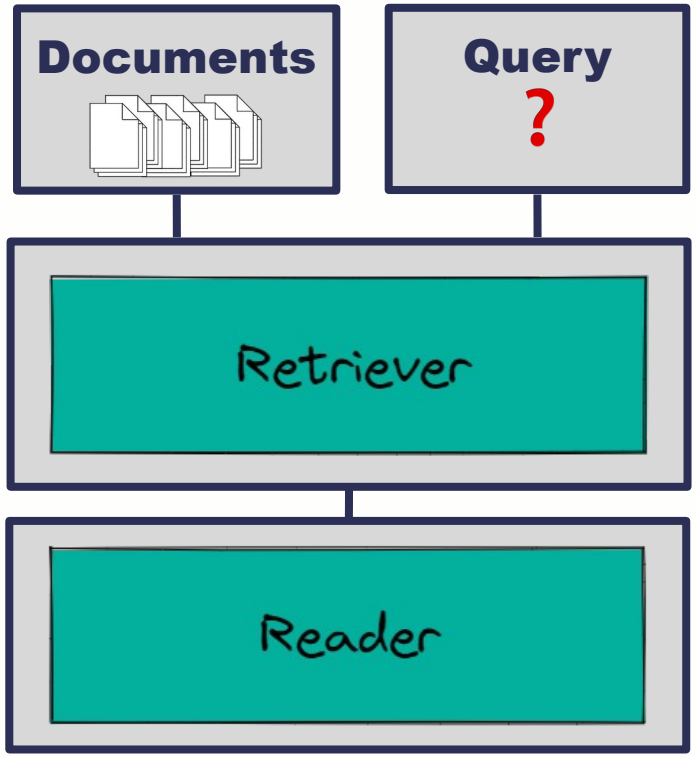


```
pip install farm-haystack
```

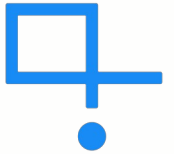


```
from haystack.nodes.retriever import EmbeddingRetriever
from haystack.nodes.reader import FARMReader

# Component to retrieve documents relevant to a query
retriever = EmbeddingRetriever(document_store=document_store, embedding_model='some_model/name')

# Component to extract answers to a query from documents
reader = FARMReader(model_name_or_path='some_other_model/name')
```

```
from haystack import Pipeline

pipeline = Pipeline()

pipeline.add_node(component=retriever, name="Retriever", inputs=["Query"])


pipeline.add_node(component=reader, name="Reader", inputs=["Retriever"])
```



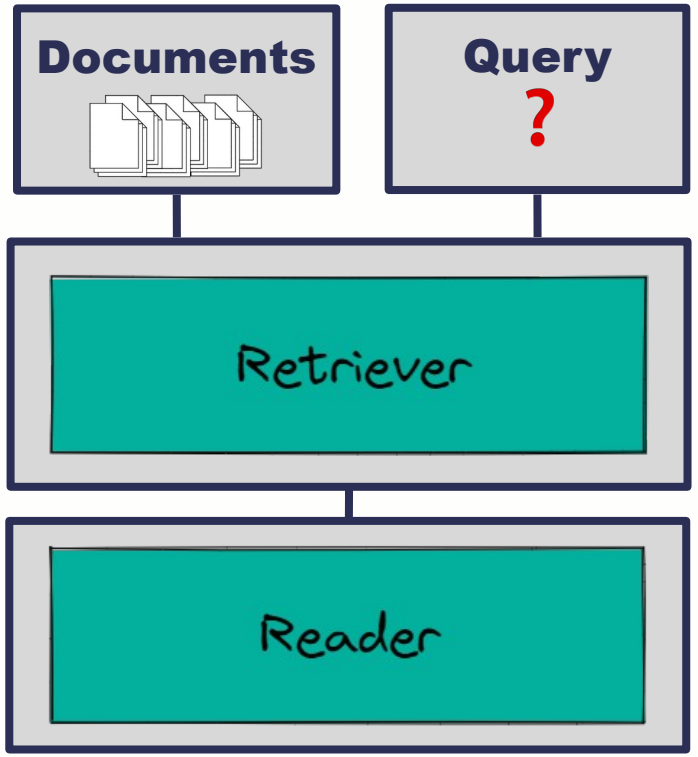
```
pip install farm-haystack
```



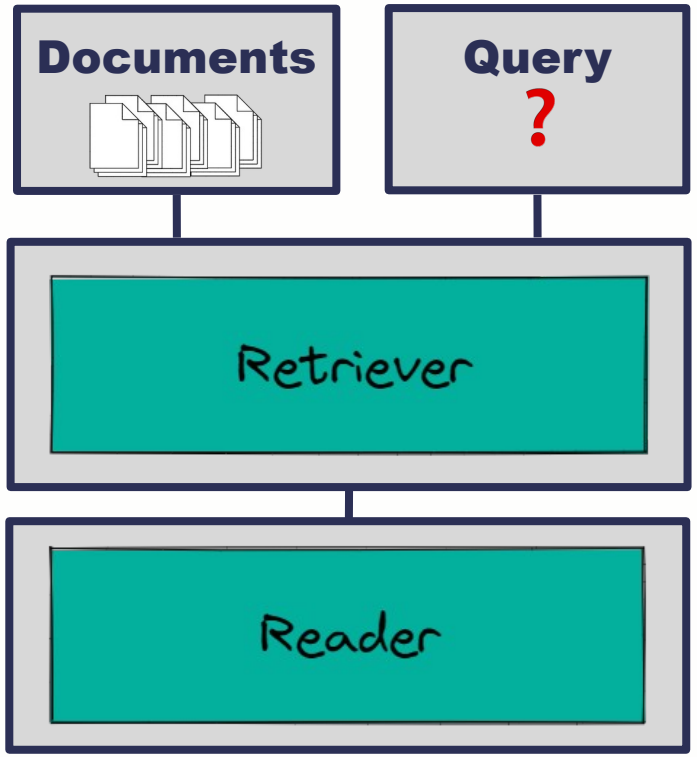
```
from haystack.nodes.retriever import EmbeddingRetriever
from haystack.nodes.reader import FARMReader

# Component to retrieve documents relevant to a query
retriever = EmbeddingRetriever(document_store=document_store, embedding_model='some_model/name')

# Component to extract answers to a query from documents
reader = FARMReader(model_name_or_path='some_other_model/name')
```

```
from haystack import Pipeline

pipeline = Pipeline()

pipeline.add_node(component=retriever, name="Retriever", inputs=["Query"])


pipeline.add_node(component=reader, name="Reader", inputs=["Retriever"])
```

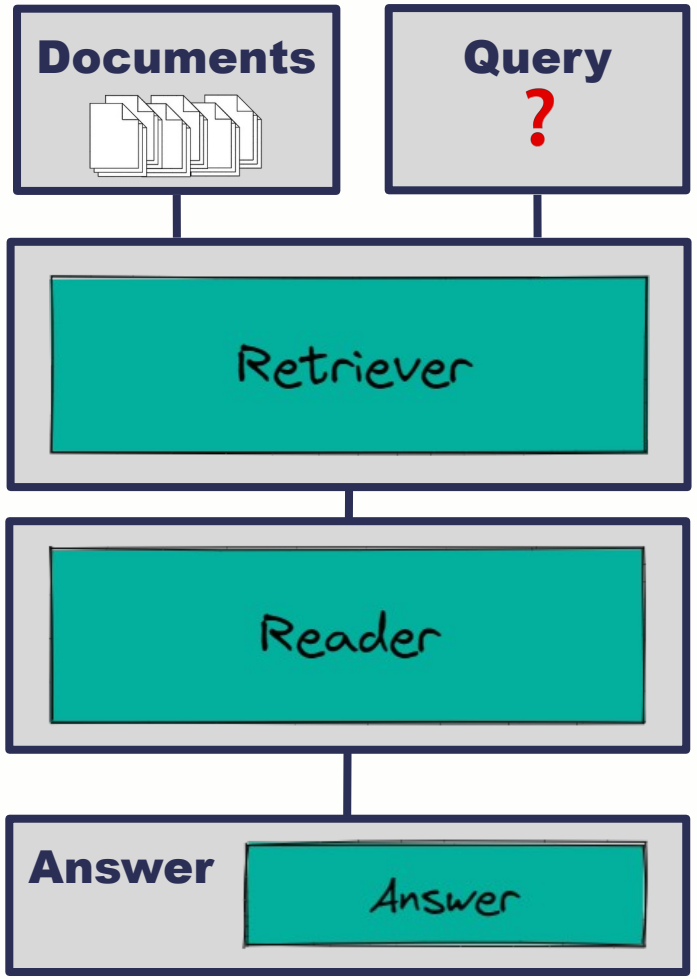
```
from haystack.pipelines import ExtractiveQAPipeline

pipeline = ExtractiveQAPipeline(retriever=retreiver, reader=reader)
```



```
pipeline.run(query="When is Milos flying to Frankfurt?")
```



```
pipeline.run(query="When is Milos flying to Frankfurt?")
```

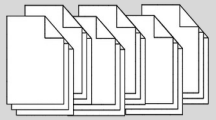


haystack
by deepset

Let's set the scene:

- **We have some data**
- **They're all about Game of Thrones**
- **We want to do Questions Answering**

Document store



We have a `DocumentStore` full of Game of Thrones data

- Specifically, we have a **`FAISSDocumentStore`**
- We have .txt files about GoT in there

Retriever

We want to retrieve the most relevant information and pass only that to the reader

- We will use the **`EmbeddingRetriever:`**
  - This will allow us to generate vector representations for our documents

Reader

Once we have candidate documents, we want to extract an answer

- We will use the **`FARMReader:`**
  - This will allow us to use an extractive QA model from Hugging Face

haystack
by deepset

Some Examples with Haystack: A simple Extractive QA pipeline

```python
from haystack import Pipeline
from haystack.document_stores import FAISSDocumentStore
from haystack.nodes import FARMReader, EmbeddingRetriever

document_store = FAISSDocumentStore()

retriever = EmbeddingRetiever(document_store=document_store, embedding_model='sentence-transformers/all-MiniLM-L6-v2')
```

haystack
by deepset

Some Examples with Haystack: A simple Extractive QA pipeline

```python
from haystack import Pipeline
from haystack.document_stores import FAISSDocumentStore
from haystack.nodes import FARMReader, EmbeddingRetriever

document_store = FAISSDocumentStore()

retriever = EmbeddingRetiever(document_store=document_store, embedding_model='sentence-transformers/all-MiniLM-L6-v2')


document_store.update_embeddings(retriever=retriever)
```

haystack
by deepset

Some Examples with Haystack: A simple Extractive QA pipeline

```python
from haystack import Pipeline
from haystack.document_stores import FAISSDocumentStore
from haystack.nodes import FARMReader, EmbeddingRetriever

document_store = FAISSDocumentStore()

retriever = EmbeddingRetiever(document_store=document_store, embedding_model='sentence-transformers/all-MiniLM-L6-v2')


document_store.update_embeddings(retriever=retriever)


reader = FARMReader(model_name_or_path='deepset/roberta-base-squad2')
```

haystack
by deepset

Some Examples with Haystack: A simple Extractive QA pipeline

```python
from haystack import Pipeline
from haystack.document_stores import FAISSDocumentStore
from haystack.nodes import FARMReader, EmbeddingRetriever

document_store = FAISSDocumentStore()

retriever = EmbeddingRetiever(document_store=document_store, embedding_model='sentence-transformers/all-MiniLM-L6-v2')


document_store.update_embeddings(retriever=retriever)


reader = FARMReader(model_name_or_path='deepset/roberta-base-squad2')


query_pipeline = Pipeline()

query_pipeline.add_node(component=retriever, name="Retriever", inputs=["Query"])
query_pipeline.add_node(component=reader, name="Reader", inputs=["Retriever"])

query_pipeline.run("Who is the father of Arya Stark?")
```

haystack by deepset

Answers in Haystack

```
[   {   'answer': 'Eddard',
        'context': 's 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 ha"},
    {   'answer': 'Balon Greyjoy',
        'context': 'He sends Theon to the Iron Islands hoping to broker an '
                   "alliance with Balon Greyjoy, Theon's father. In exchange "
                   'for Greyjoy support, Robb as the King '},
    {   'answer': 'Brynden Tully',
        'context': 'o the weather. Sandor decides to instead take her to her '
                   'great-uncle Brynden Tully. On their way to Riverrun, they '
                   "encounter two men on Arya's death l"}]
```

Now let’s generate some human-like answers

haystack by deepset

Some Examples with Haystack: Answer Generation (with models from OpenAI)

```
from haystack.nodes import PromptNode

answer_generator = PromptNode(model_name_or_path='text-davinci-003', api_key='OPEN_AI_API_KEY')

answer_generator("When is Milos flying to Frankfurt?")
```

haystack
by deepset

Some Examples with Haystack: Answer Generation (with models from OpenAI)

```
from haystack.nodes import PromptNode

answer_generator = PromptNode(model_name_or_path='text-davinci-003', api_key='OPEN_AI_API_KEY')

answer_generator("When is Milos flying to Frankfurt?")

>> ["Milos's flight to Frankfurt is scheduled for August 7, 2020."]
```

haystack
by deepset

Some Examples with Haystack: Answer Generation (with models from OpenAI)

```
from haystack.nodes import PromptNode

answer_generator = PromptNode(model_name_or_path='text-davinci-003', api_key='OPEN_AI_API_KEY')

answer_generator("When is Milos flying to Frankfurt?")

>> ["Milos's flight to Frankfurt is scheduled for August 7, 2020."]


>> ['It is not possible to answer this question without more information.']
```

haystack
by deepset

Some Examples with Haystack: Answer Generation (with models from OpenAI)

```
from haystack.nodes import PromptNode

answer_generator = PromptNode(model_name_or_path='text-davinci-003', api_key='OPEN_AI_API_KEY')

answer_generator("When is Milos flying to Frankfurt?")

>> ["Milos's flight to Frankfurt is scheduled for August 7, 2020."]


>> ['It is not possible to answer this question without more information.']

answer_generator("When is Milos travelling somewhere?")
```

haystack
by deepset

Some Examples with Haystack: Answer Generation (with models from OpenAI)

```
from haystack.nodes import PromptNode

answer_generator = PromptNode(model_name_or_path='text-davinci-003', api_key='OPEN_AI_API_KEY')

answer_generator("When is Milos flying to Frankfurt?")

>> ["Milos's flight to Frankfurt is scheduled for August 7, 2020."]


>> ['It is not possible to answer this question without more information.']

answer_generator("When is Milos travelling somewhere?")

>> ['Milos is travelling when he has the time and money available to do so.']
```

haystack
by deepset

Some Examples with Haystack: Answer Generation (with models from OpenAI)

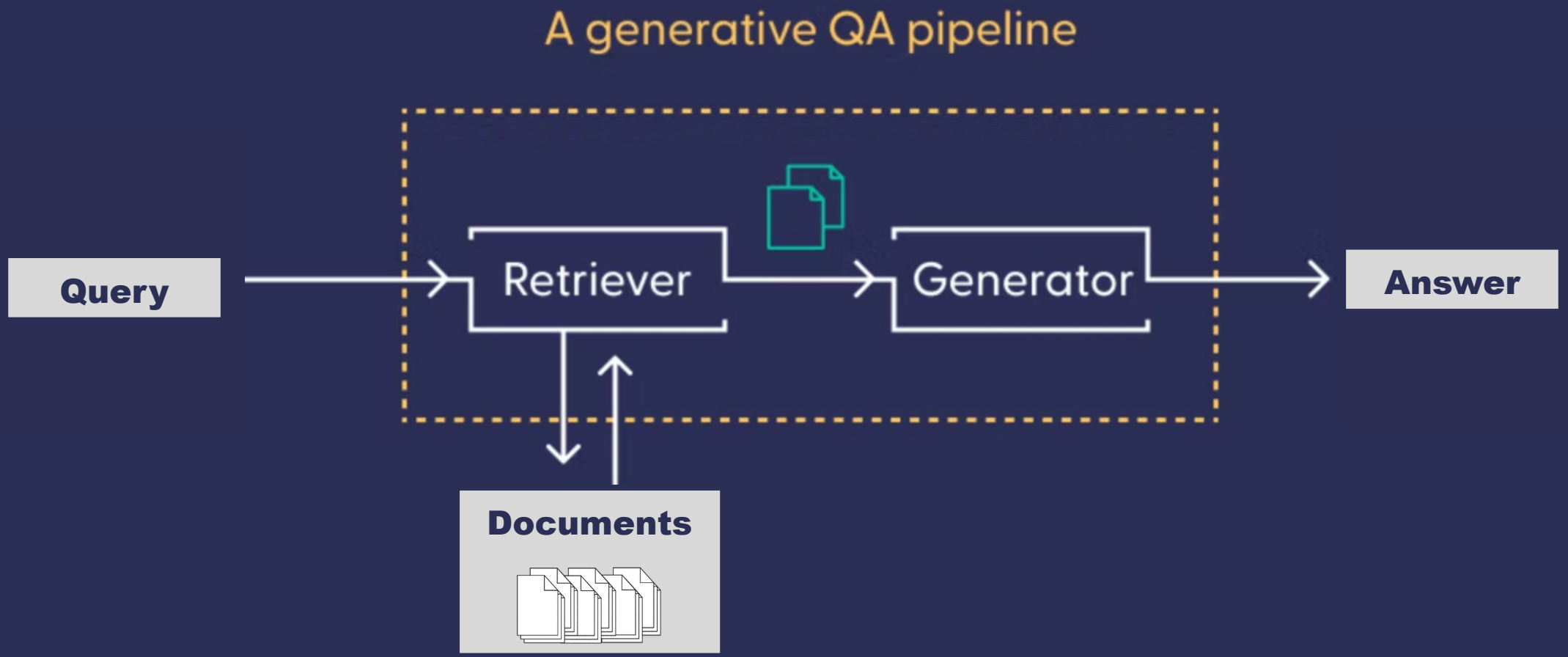
```
from haystack.nodes import PromptNode

answer_generator = PromptNode(model_name_or_path='text-davinci-003', api_key='OPEN_AI_API_KEY')

answer_generator("When is Milos flying to Frankfurt?")

>> ["Milos's flight to Frankfurt is scheduled for August 7, 2020."]


>> ['It is not possible to answer this question without more information.']

answer_generator("When is Milos travelling somewhere?")

>> ['Milos is travelling when he has the time and money available to do so.']

answer_generator("Who is Milos?")
```

haystack by deepset

Some Examples with Haystack: Answer Generation (with models from OpenAI)

```
from haystack.nodes import PromptNode

answer_generator = PromptNode(model_name_or_path='text-davinci-003', api_key='OPEN_AI_API_KEY')

answer_generator("When is Milos flying to Frankfurt?")

>> ["Milos's flight to Frankfurt is scheduled for August 7, 2020."]


>> ['It is not possible to answer this question without more information.']

answer_generator("When is Milos travelling somewhere?")

>> ['Milos is travelling when he has the time and money available to do so.']

answer_generator("Who is Milos?")

>> ['Milos is a Greek island in the Aegean Sea, located to the south-east of mainland Greece. It is the largest island in
the Cyclades group of islands and is known for its stunning beaches, picturesque villages, and archaeological sites.']
```

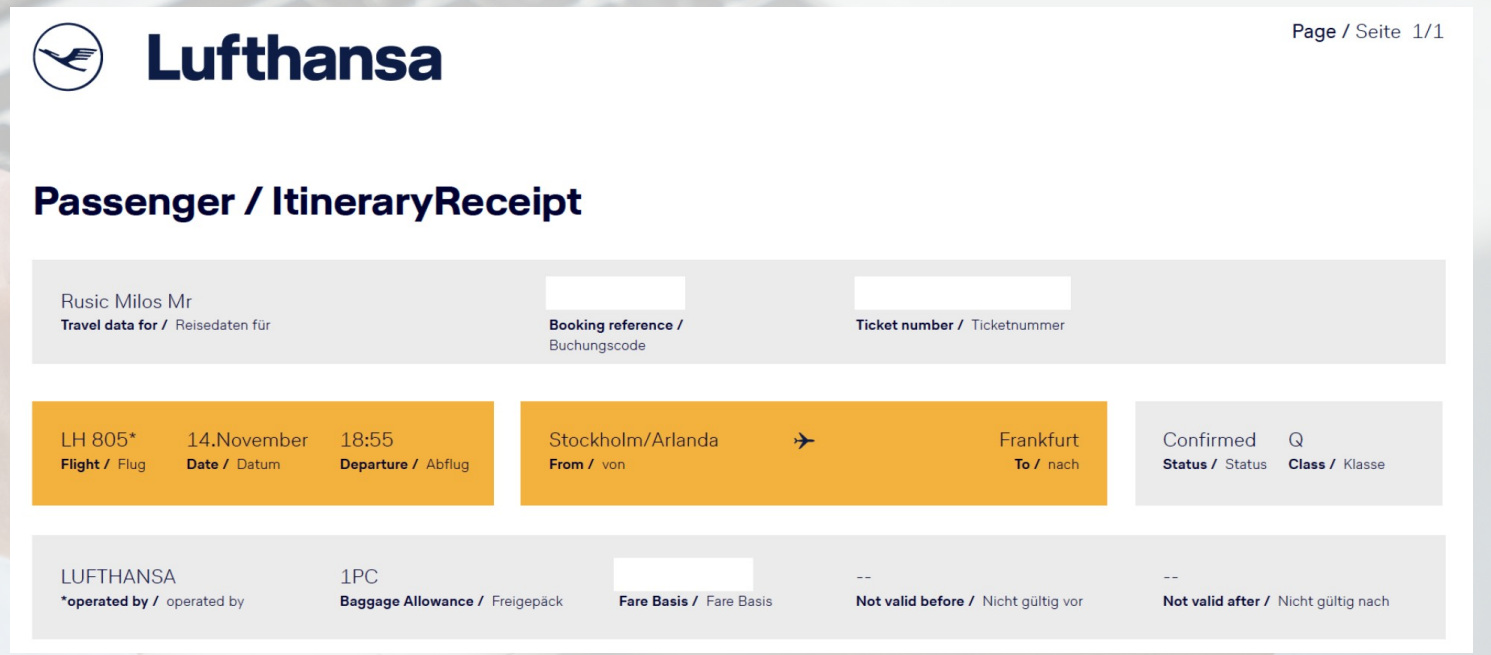
Using Generative Models On Our Own Data



haystack by deepset

Some Examples with Haystack:

Retrieval Augmented Answer Generation

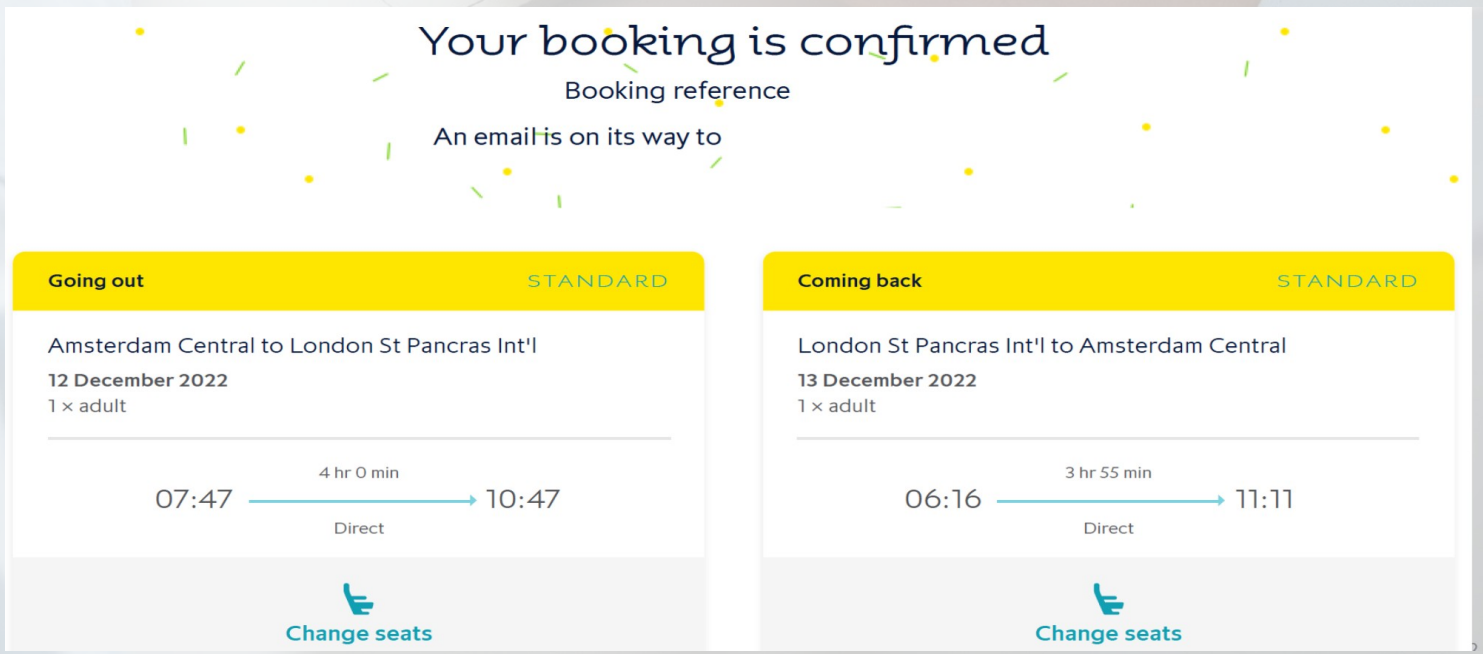


Lufthansa

Page / Seite 1/1

Passenger / ItineraryReceipt

Rusic Milos Mr **Travel data for /** Reisedaten für		**Booking reference /** Buchungscode	**Ticket number /** Ticketnummer
LH 805* **Flight /** Flug	14.November **Date /** Datum	18:55 **Departure /** Abflug	Stockholm/Arlanda **From /** von → Frankfurt **To /** nach
Confirmed **Status /** Status	Q **Class /** Klasse		
LUFTHANSA ***operated by /** operated by	1PC **Baggage Allowance /** Freigepäck	**Fare Basis /** Fare Basis	-- **Not valid before /** Nicht gültig vor	-- **Not valid after /** Nicht gültig nach



Your booking is confirmed

Booking reference

An email is on its way to

Going out STANDARD

Amsterdam Central to London St Pancras Int'l

12 December 2022

1 × adult

07:47 — 4 hr 0 min — Direct — 10:47

Change seats

Coming back STANDARD

London St Pancras Int'l to Amsterdam Central

13 December 2022

1 × adult

06:16 — 3 hr 55 min — Direct — 11:11

Change seats

Some Examples with Haystack: Retrieval Augmented Answer Generation

```
from haystack.document_stores import FAISSDocumentStore
from haystack.nodes import EmbeddingRetriever, PDFToTextConverter, OpenAIAnswerGenerator, PreProcessor

document_store = FAISSDocumentStore(faiss_index_factory_str="Flat", embedding_dim=1536)

retriever = EmbeddingRetriever(document_store= document_store,
                               embedding_model='text-embedding-ada-002',
                               batch_size = 32,
                               api_key = 'YOUR_OPENAI_KEY',)
converter = PDFToTextConverter()

preprocessor = PreProcessor(split_by = 'word',
                            split_length = 250,
                            split_overlap = 20,
                            split_respect_sentence_boundary = True)
```

haystack by deepset

Some Examples with Haystack: Retrieval Augmented Answer Generation

```python
from haystack.pipelines import GenerativeQAPipeline

my_search_engine = GenerativeQAPipeline(generator=generator, retriever=retriever)
```

```python
from haystack import Pipeline

indexing_pipeline = Pipeline()

indexing_pipeline.add_node(component=converter, name="PDFConverter", inputs=["File"])
indexing_pipeline.add_node(component=preprocessor, name="PreProcessor", inputs=["PDFConverter"])
indexing_pipeline.add_node(component=retriever, name="Retriever", inputs=["PreProcessor"])
indexing_pipeline.add_node(component=document_store, name="DocumentStore", inputs=["Retriever"])

indexing_pipeline.run(file_paths=['milos_ticket.pdf', 'tuana_ticket.pdf'])

query_pipeline = Pipeline()

query_pipeline.add_node(component=retriever, name="Retriever", inputs=['Query'])
query_pipeline.add_node(component=generator, name="AnswerGenerator", inputs=['Retriever'])
```

haystack by deepset

Some Examples with Haystack: Retrieval Augmented Answer Generation

```
query_pipeline.run(query='Who is Milos?')

>> 'Milos is the passenger whose travel data is on the Passenger/Itinerary
Receipt.'

query_pipeline.run(query='When is Milos flying to Frankfurt?')

>> 'Milos is flying to Frankfurt on 14 November at 18:55.

query_pipeline.run(query='Who is travelling to London?')

>> 'Tuana Celik is travelling to London.'
```

Conclusion

- Extractive models are great at retrieving knowledge from some context
- Generative models are cool and they provide human-like answers
- Combining a retrieval augmentation step in generative pipelines can let you effectively use these models
- Haystack is an open source framework built in Python and it’s extended every day to cover more and more NLP tasks
    - Contributions welcome :)

haystack
by deepset

Where to find us

- We have an open Discord server
- We have regular meetups! Go check out the **Open NLP Meetup**
- GitHub: Haystack is open source ⭐
- Twitter @deepset_ai

Try out your first QA pipeline with our tutorial:

Thank you!

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

Haystack Tutorials
https://haystack.deepset.ai/tutorials

Discord Server
https://haystack.deepset.ai/community

@deepset_ai

deepset

@tuanacelik