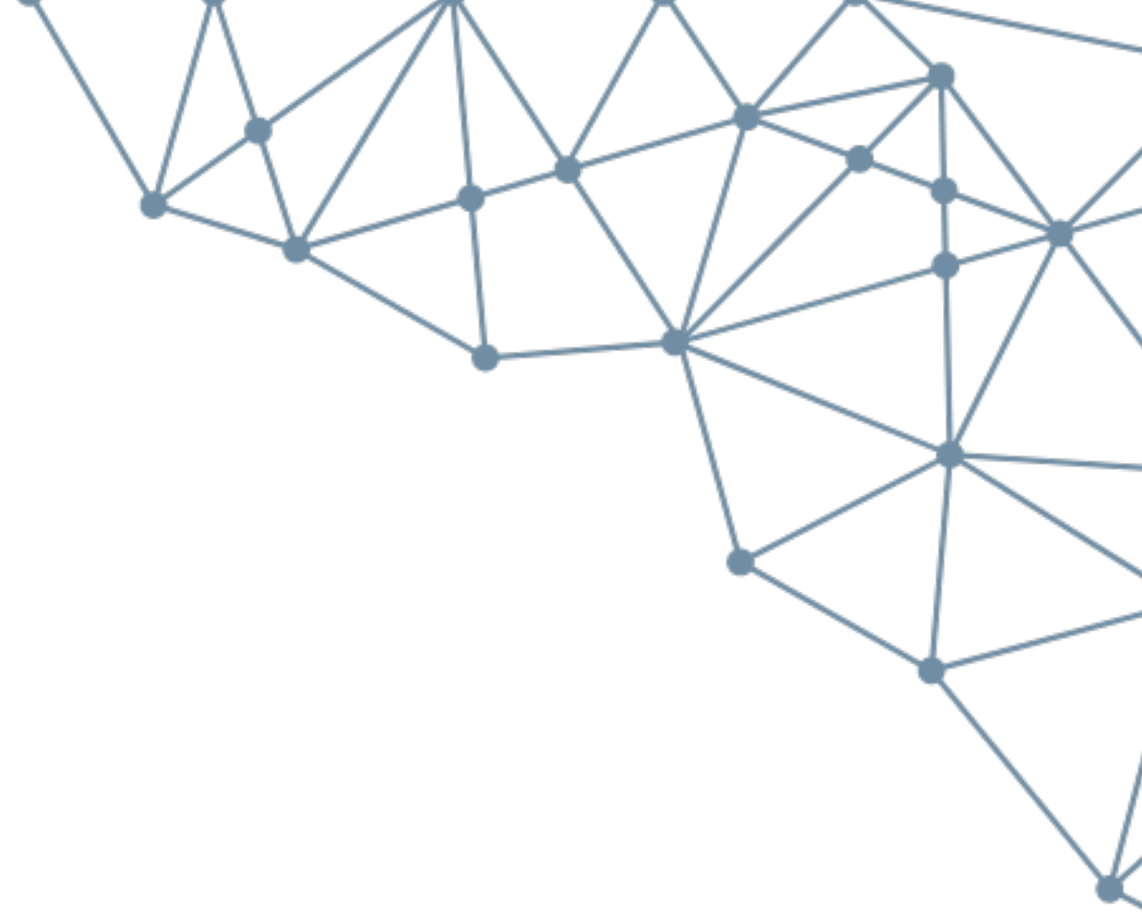
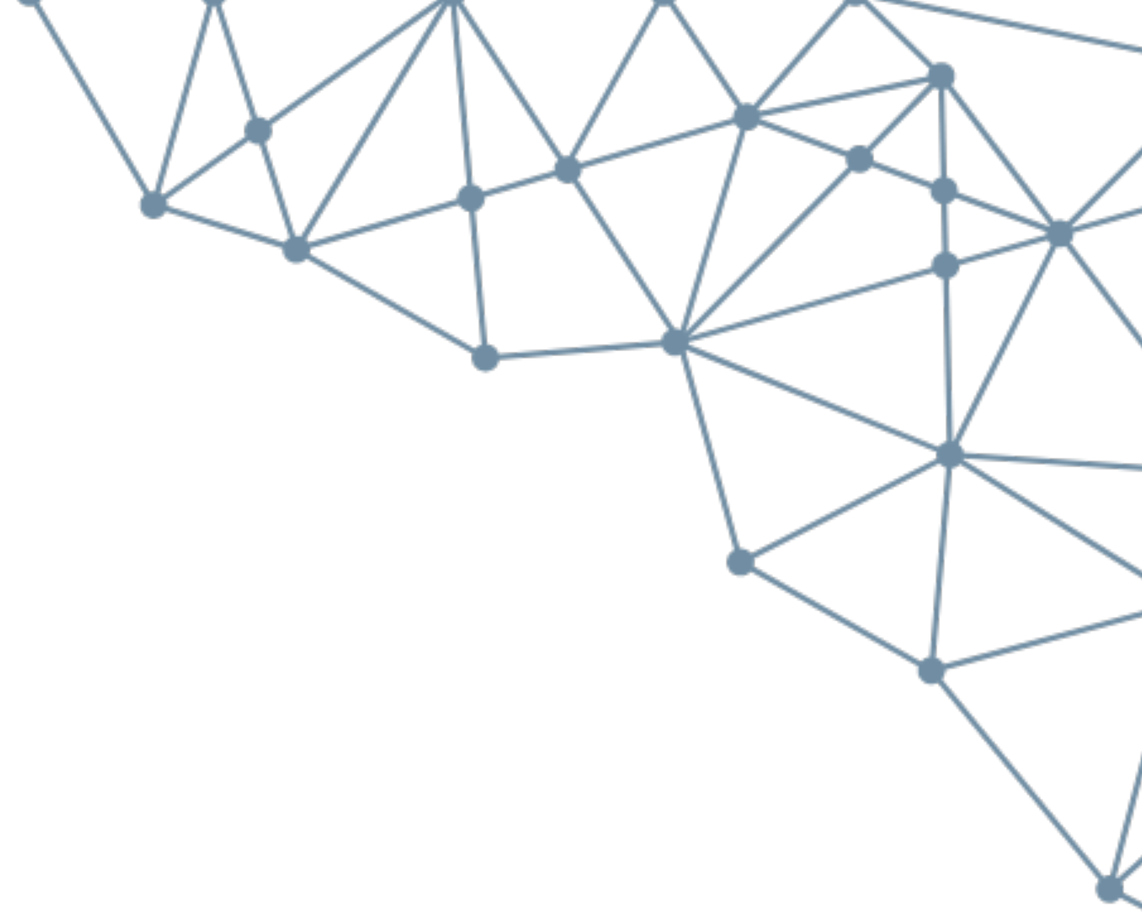


ABB - Session 3

RAG, Text Embeddings

Shaw Talebi

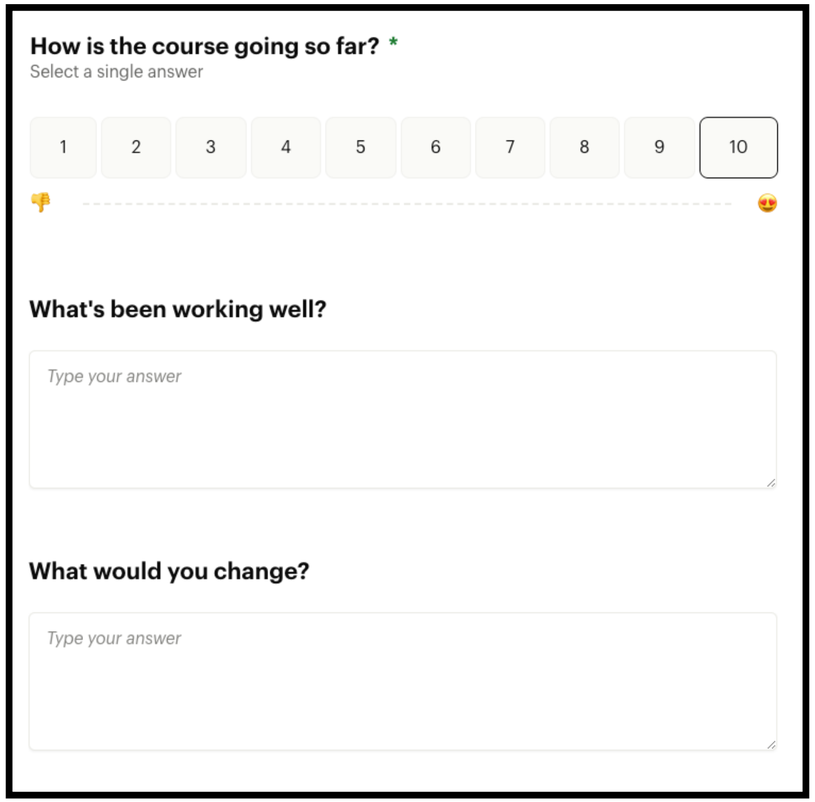
Today's Session



1. **Housekeeping**
    1.1. Announcements
    1.2. Homework 2
2. **RAG**
    2.1. What is RAG?
    2.2. RAG Implementation
    2.3. Text Embeddings
    2.4. Two Use Cases
3. **Examples**
    3.1. Analyzing Survey Data with Embeddings
    3.2. RAG with LlamaIndex
    3.3. PDF Parsing with Docling

Announcements

1) Mid-course Survey



2) Office Hours

Mondays, 4 - 5PM CST

or

Mondays, 4 - 4:30PM CST
+
Wednesdays, 12 - 12:30PM CST

Homework 2

Shoutouts 🎉

Review Sentiment Analysis
Prabhakar Somu

Ad Break Detector
Jerry Bonner

Tech Document Summarizer
Chris Gervais

LLM Invoice Parser
Thomas Helms

2 Levels of LLM Development

How to get LLMs to do what you want…

Level 1

Adapting models via prompts and tools

Prompt Engineering 

RAG 

Tool-use 

Level 2

Adapting models via additional training

Fine-tuning 

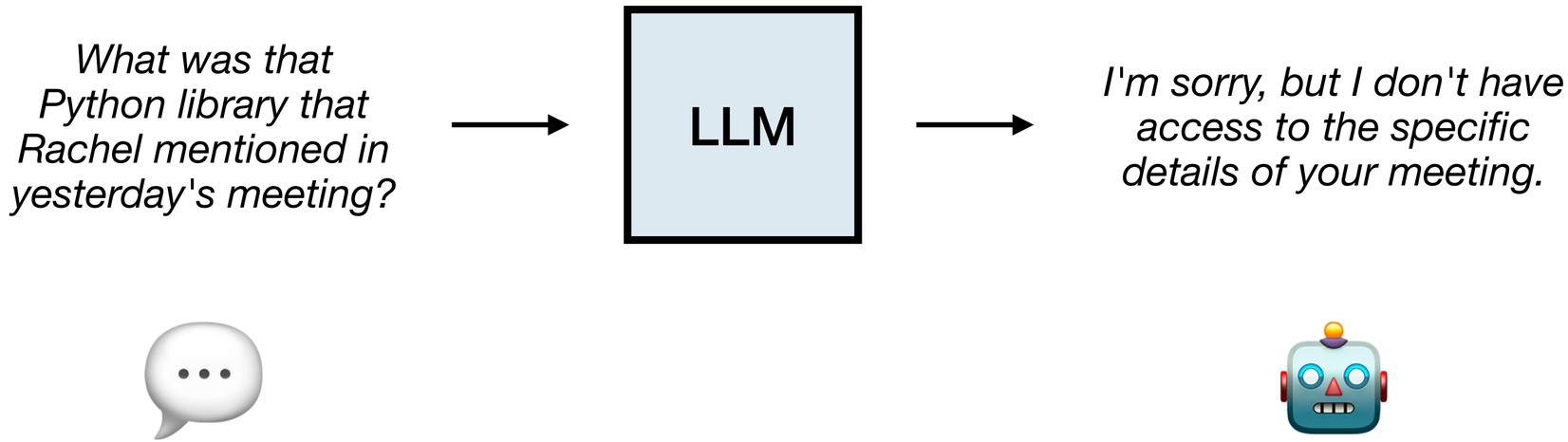
Post-training 

RAG

What is RAG?

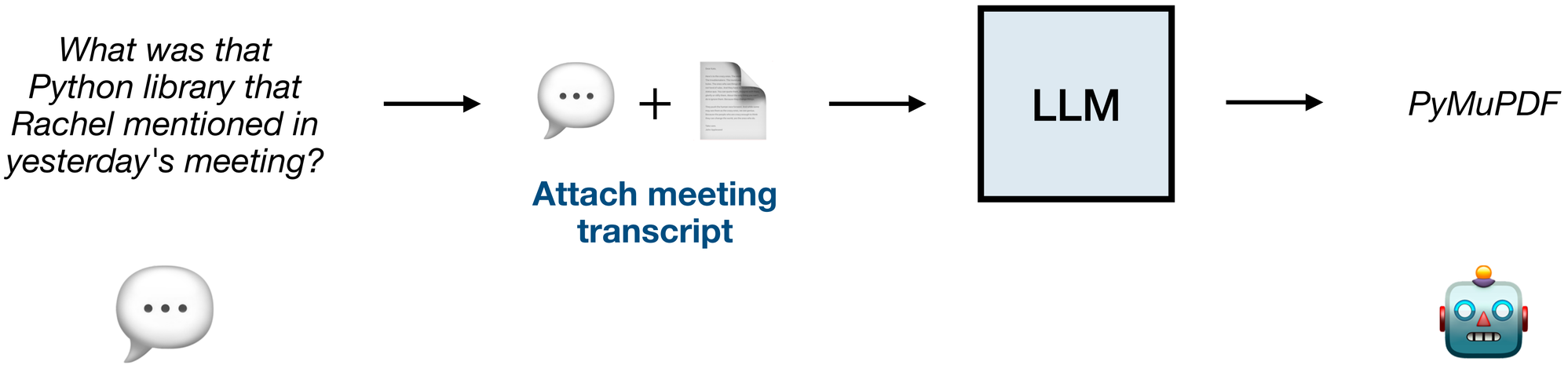
Improving an LLM's responses by automatically providing it relevant context

RAG = Retrieval Augmented Generation



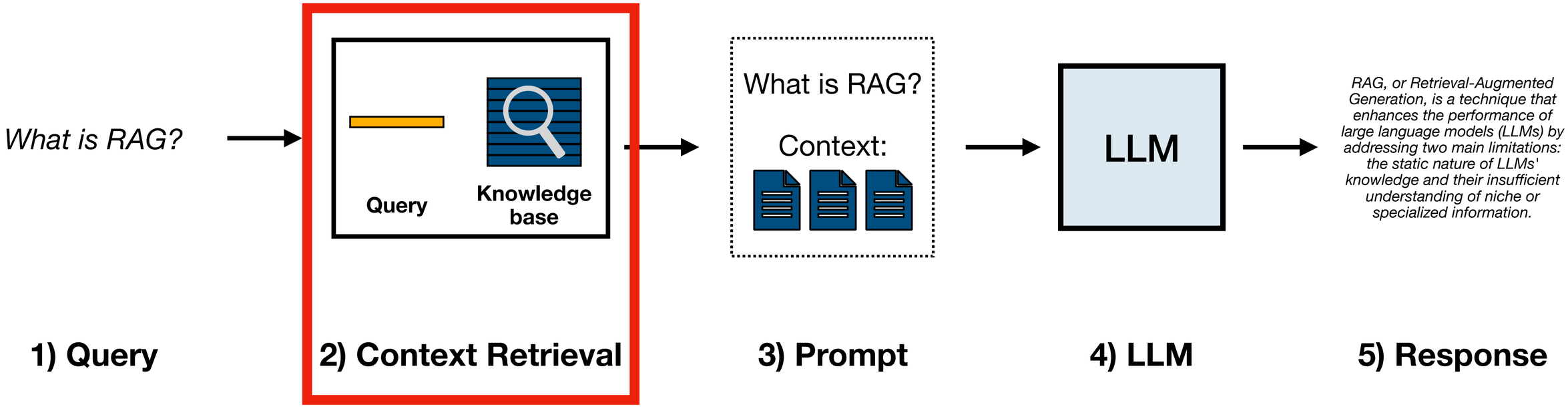
What is RAG?

Improving an LLM's responses by automatically providing it relevant context



RAG Implementation

5-step process

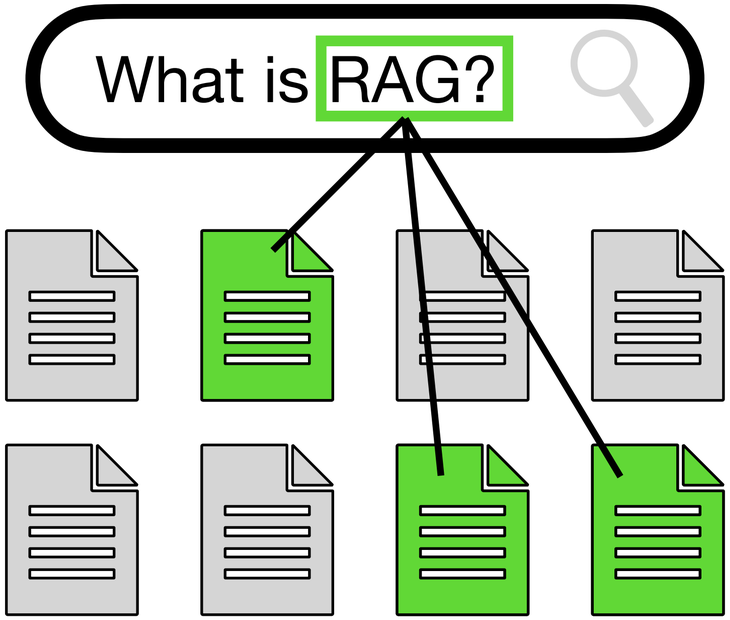


[1]

2 Ways Retrieval Approaches

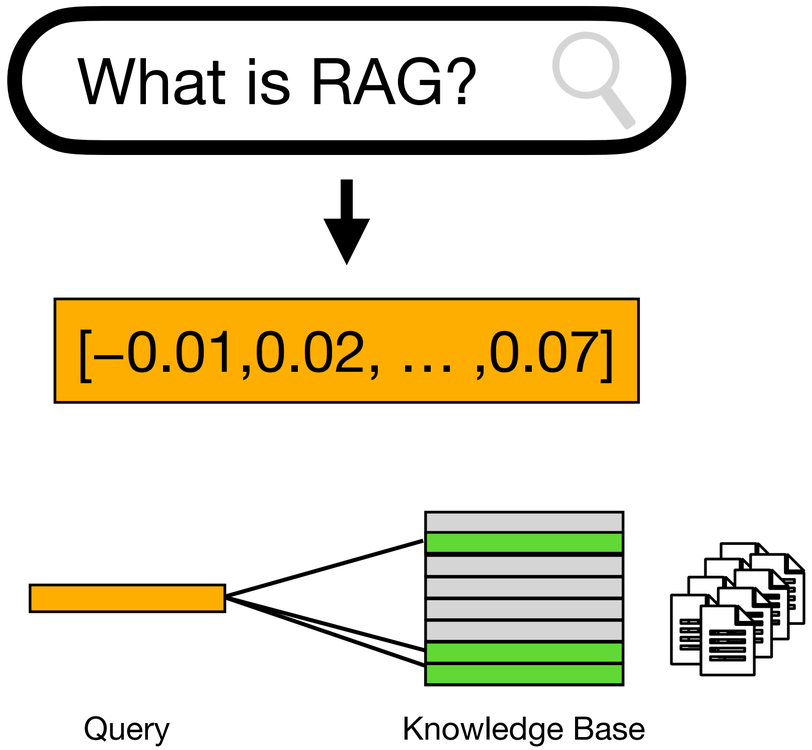
Way 1

Keyword-based Search



Way 2

Vector Search



[3]

Text Embeddings

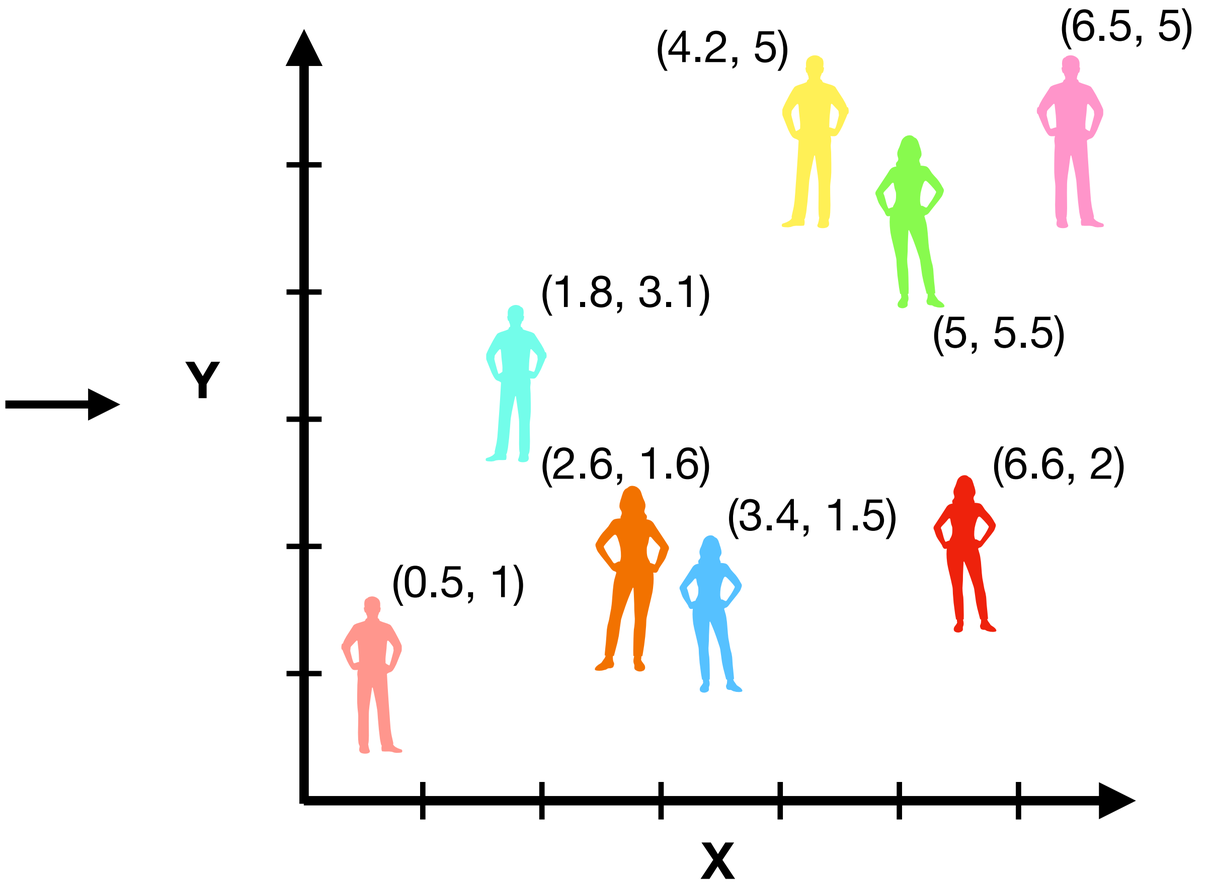
Translate words into numbers

Job Description		Text Embedding
Data Analyst in retail, 5 yrs	→	(3.4, 1.5)
ML Engineer in fintech, 3 yrs	→	(1.8, 3.1)
Data Scientist in healthcare, 10+ yrs.	→	(6.6, 2)
Database Admin for e-commerce, 7 yrs.	→	(4.2, 5)
BI Analyst in hospitality, early career.	→	(0.5, 1)
Data Architect, 15 yrs.	→	(6.5, 5)
Freelance Data Visualization Specialist, 4 yrs.	→	(2.6, 1.6)
Senior Data Engineer in automotive, 8 yrs.	→	(5, 5.5)

Text Embeddings

Translate words into ***meaningful*** numbers

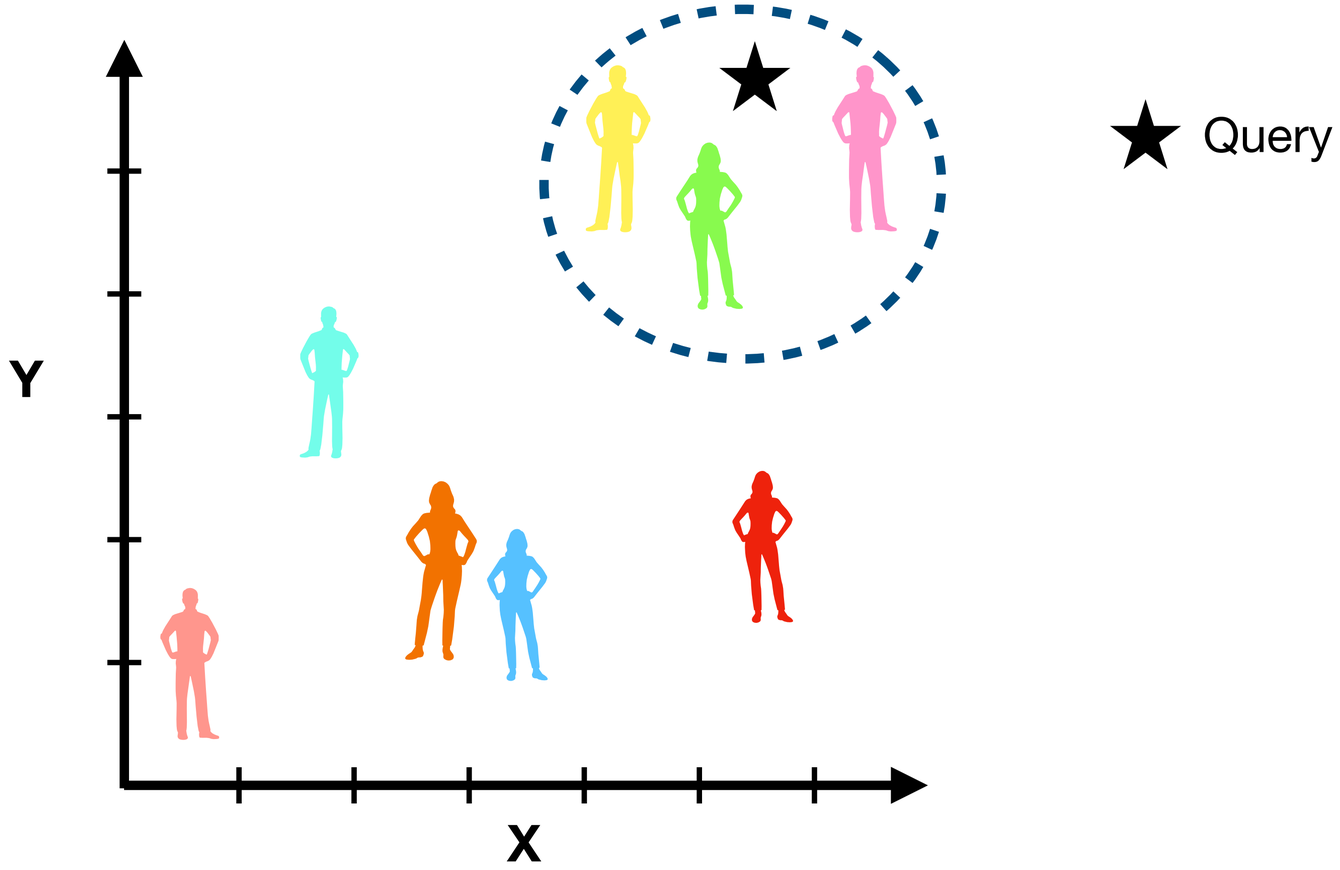
Job Description
Data Analyst in retail, 5 yrs
ML Engineer in fintech, 3 yrs
Data Scientist in healthcare, 10+ yrs.
Database Admin for e-commerce, 7 yrs.
BI Analyst in hospitality, early career.
Data Architect, 15 yrs.
Freelance Data Visualization Specialist, 4 yrs.
Senior Data Engineer in automotive, 8 yrs.



Semantic Search

Return results based on the *meaning* of user's query

"I need someone to build my data infrastructure"



Computing Similarities

Cosine similarity

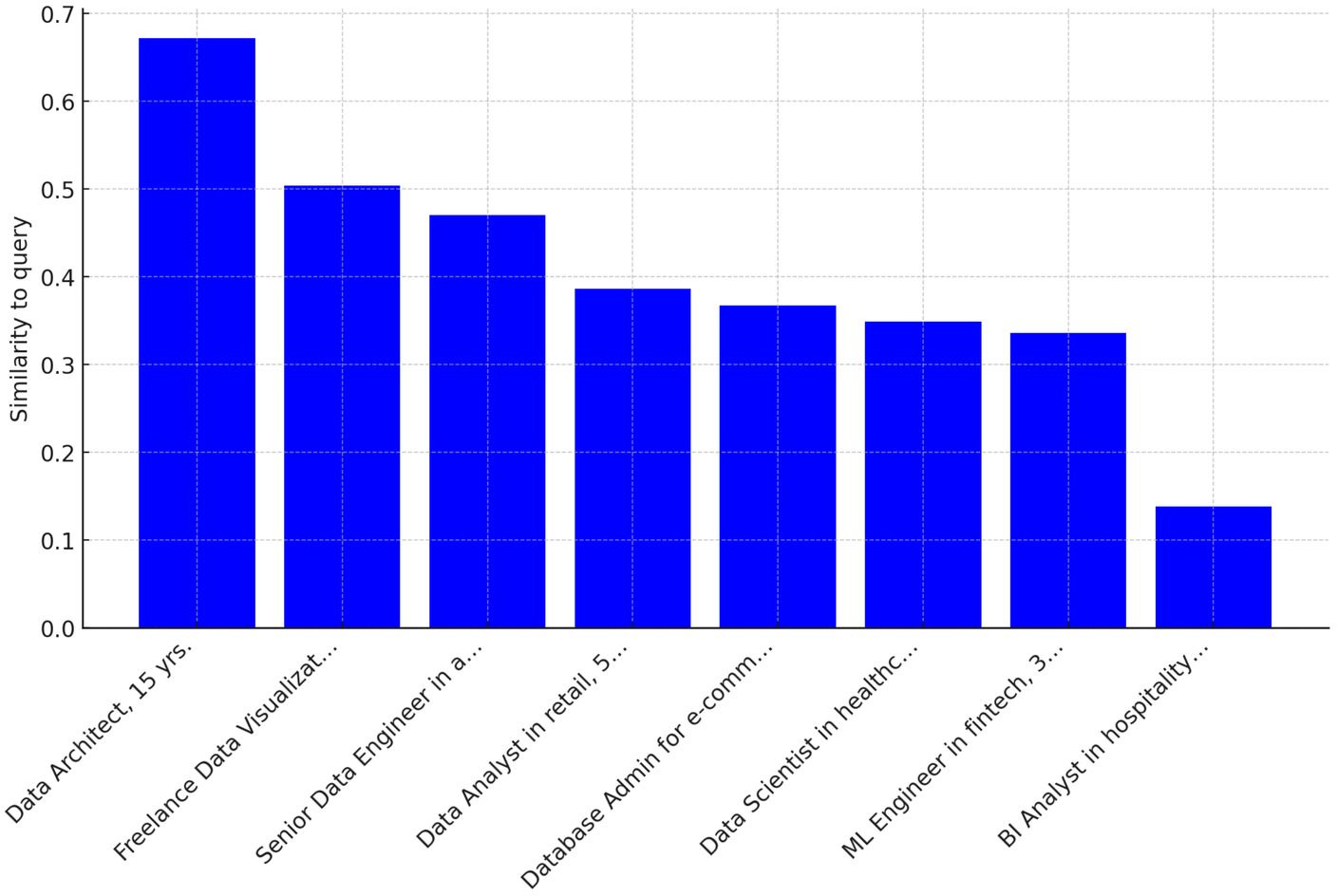
$\vec{q}$ = Query embedding

$\vec{d}_i = i^{th}$ JD embedding

Cosine Similarity

$$s_i = \frac{\vec{q} \cdot \vec{d}_i}{\|\vec{q}\| \|\vec{d}_i\|}$$

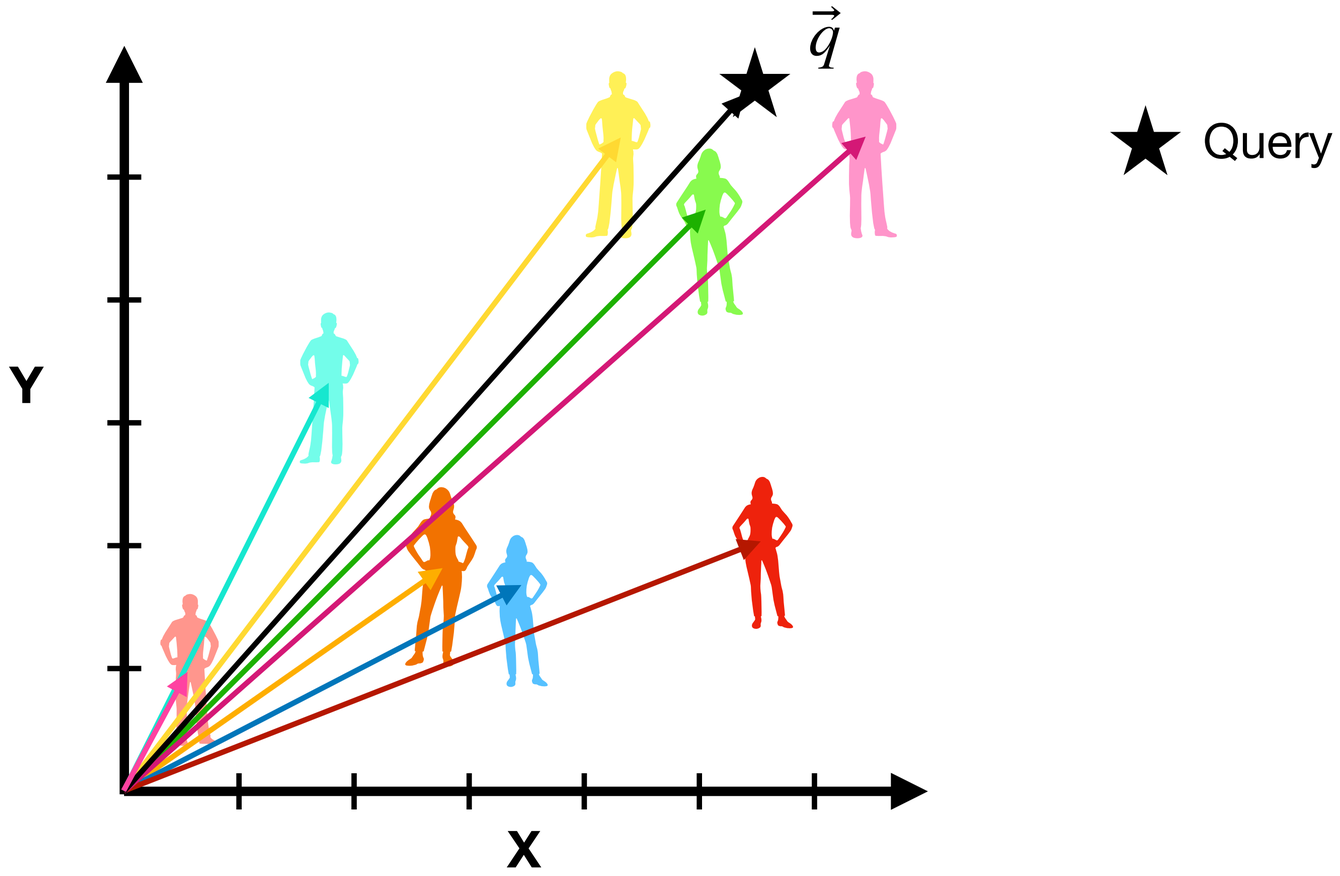
Cosine Similarities between $\vec{q}$ and $\vec{d}_i$



Visualizing Similarities

Cosine similarity

“I need someone to build my data infrastructure”

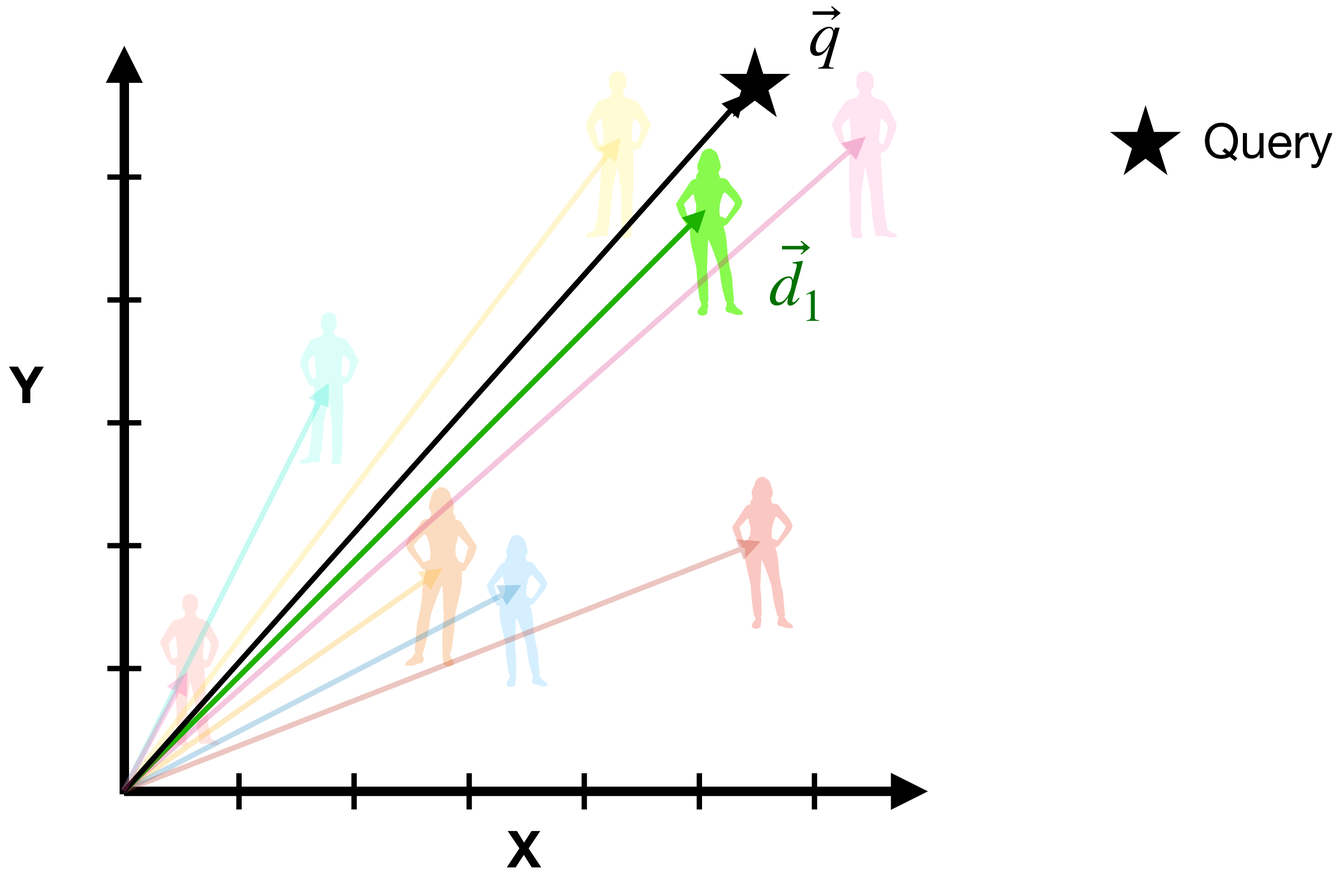


[4]

Visualizing Similarities

Cosine similarity

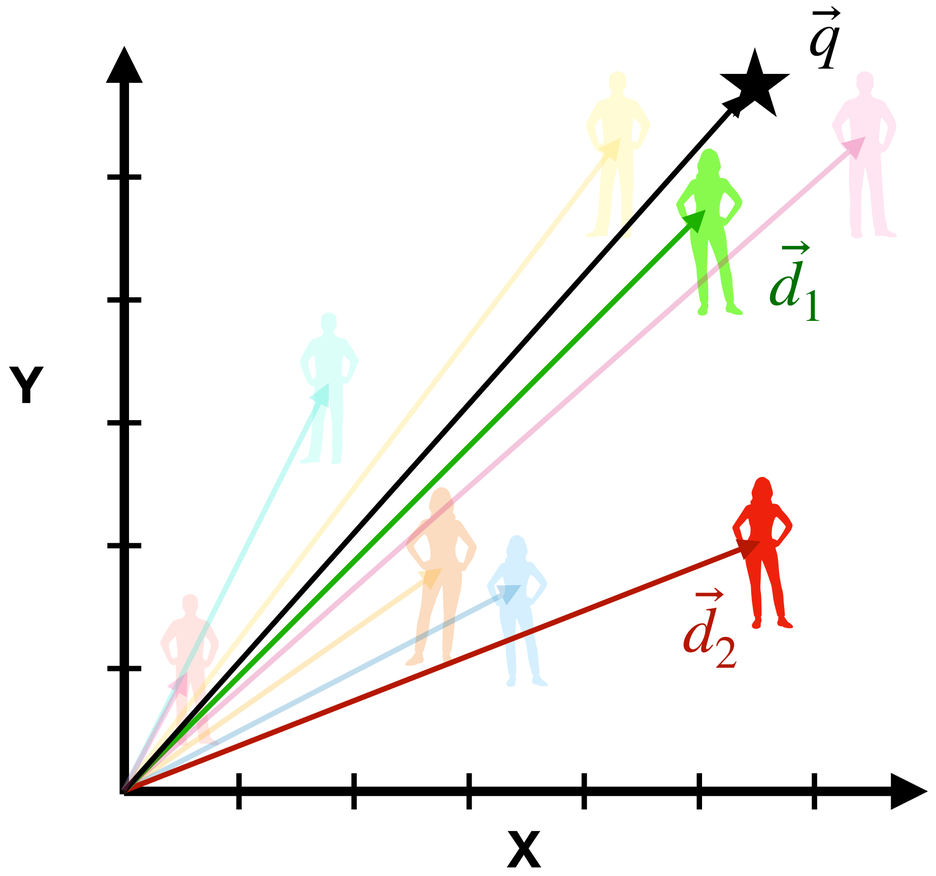
“I need someone to build my data infrastructure”



Visualizing Similarities

Cosine similarity

“I need someone to build my data infrastructure”



★ Query

Cosine Similarity

$$s_i = \frac{\vec{q} \cdot \vec{d}_i}{\|\vec{q}\| \|\vec{d}_i\|}$$

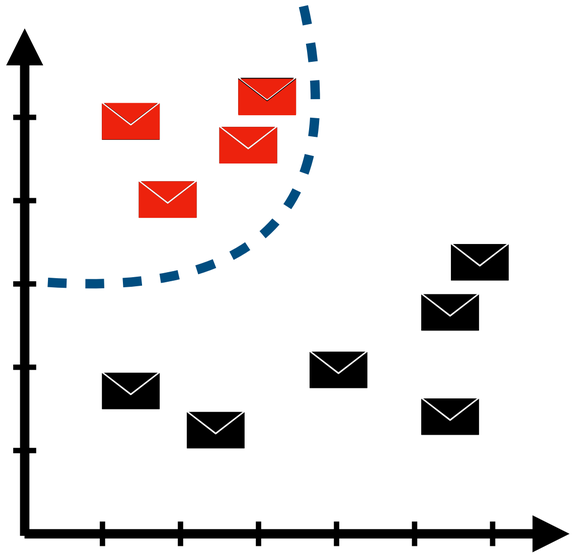
$$s_1 > s_2$$

Smaller θ $\Longrightarrow$ larger s

[4]

More Use Cases

Embeddings make text computable!

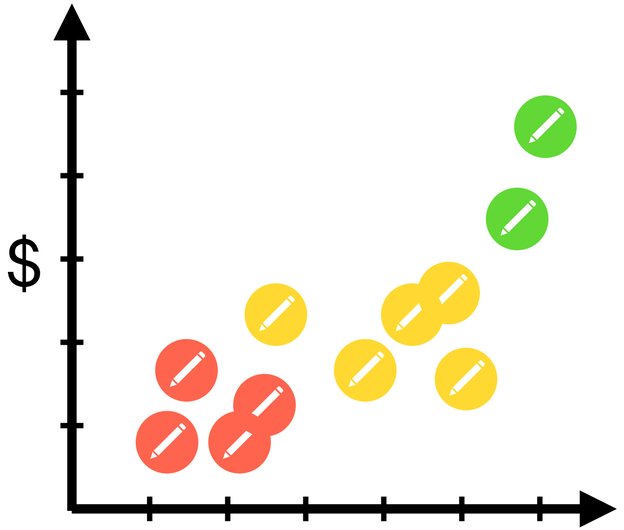


Text Classification

e.g. spam detection

Clustering

e.g. ICA analysis



Regression

e.g. product sales forecasting

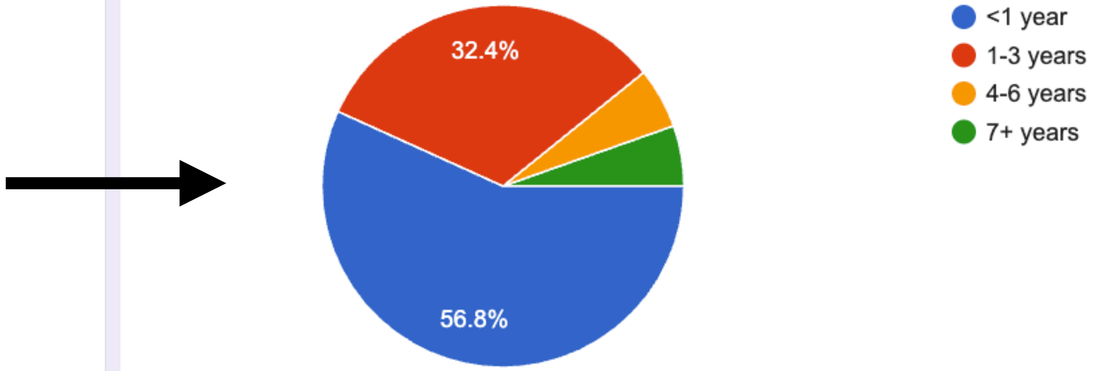
Examples

Example 1

Analyzing Survey Data with Embeddings (Motivation)

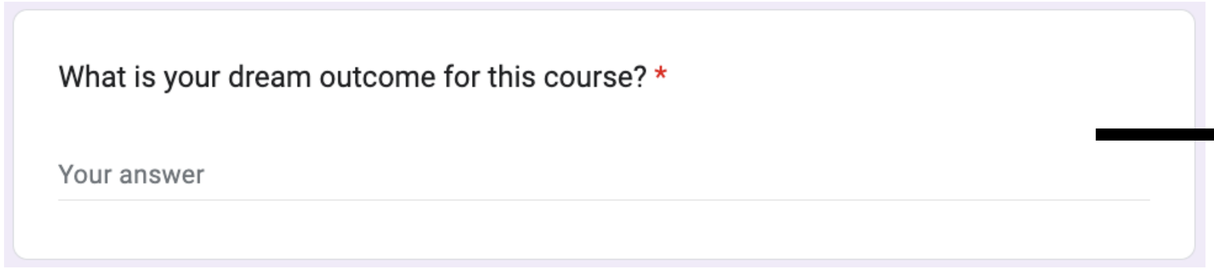
What is your current level of experience with AI/ML? *

- <1 year
- 1-3 years
- 4-6 years
- 7+ years



Example 1

Analyzing Survey Data with Embeddings (Motivation)



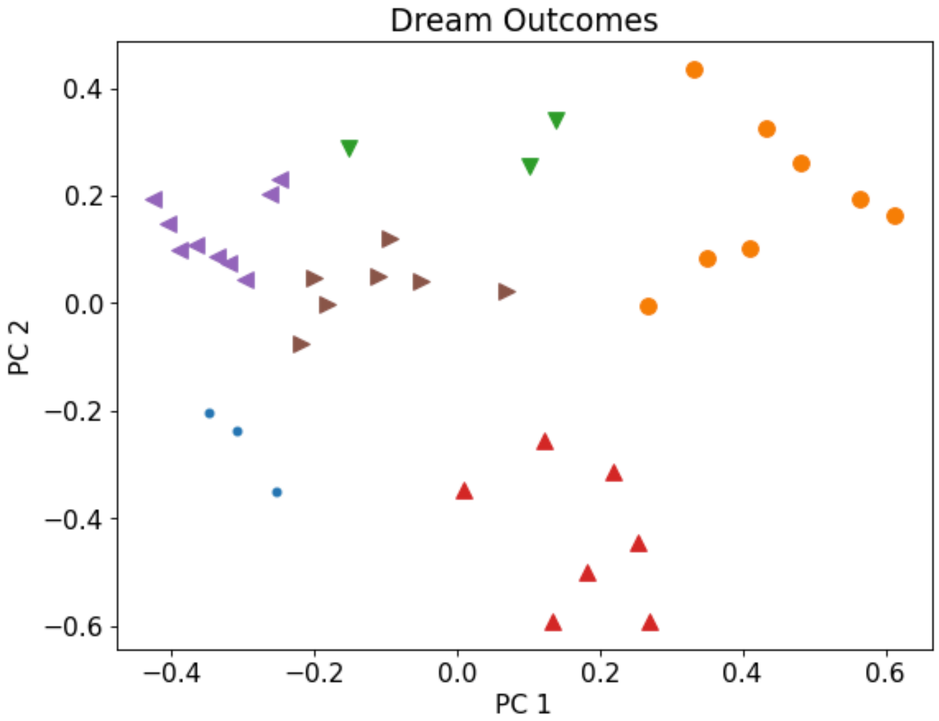
- Hands on projects, and switch to Data science career path from research
- Get to know Shaw better! Plus, have a clear path with resources to guide you on where to go and what to do for the future
- Fundamentals understanding, Hands on skills, small projects running in diff frameworks, create a small GH portfolio.
- Build multiple AI applications
- I would like to learn how to practically setup LLM application in a production environment so that I can start creating some AI web apps for internal and external use.
- Ability to train model based on a data set, and how to do predictive analysis.
- Develop a good foundation in AI/ML.
- I love to figure out good strategies to use in my software solutions using AI
- Learn to decide and advice between different data science archictures and options
- Implement practical AI cases
- Foundation in AI/ML
- Land a program manager job working with AI product/program
- Build an app
- able to create a LLM from scratch
- Get a new job
- At my former job I had a coworker who had the tedious task of making a weekly report summarizing local news for the boss. At the time I was completely sure that was something AI could do, but I didn't have the time nor the knowledge to develop such a thing. I constantly find myself having ideas like this, so a dream outcome would be finishing the course with at least a roadmap for making such an app.
- Build cool products
- Be able to be comfortable with GenAi
- A certificate to showcase my new skills, A jump start to do my own programing and be able to communicate with programmers
- Setup my own environment to compare Machine Learning statistics created by my companies Data Scientists against my own environment.
- Be able to create and deploy my own Ai powered apps
- Build and launch a product
- Getting skills to build AI technologies for many projects
- Being able to implement llms into projects
- Landing a new job
- to deepen my understanding of advanced AI techniques.And I also aim to expand my professional network and collaborate with like-minded individuals to explore new opportunities in the AI landscape.
- Be able to train a model and have some practical usage of llm.
- Learn to identify the ML solution and lead projects based on AI
- Able to put into practice AI/ML for real bunsiess solutions
- Master AI powered productivity tools to streamline regulatory compliance work
- Learn Python related to AI. Implement prototypes.
- 1) build GPT that accesses functions and APIs, 2) hands-on fine-tune a model using LoRA 3) write and debug Python code (with AI assistance) that accesses ChatGPT, Perplexity and Google search, 4) build working agent(s) that can output results in less than 10 seconds 5) possibly use a RAG efficiently for booklength PDF texts
- Fully understand AI and LLM, and how to build one using Python
- Be able to comfortably add AI into daily or business use cases
- Build my own AI product with the AI services available these days
- I want to build my own MicroSaaS products
- I love to apply ai solution to existing and future back-end projects

Objective, quantitative analysis is **challenging.**

Example 1

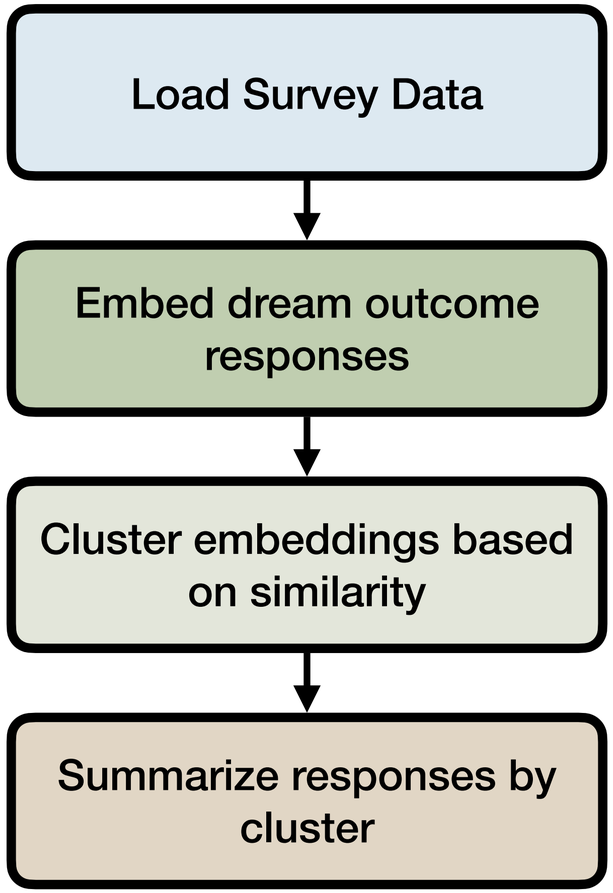
Analyzing Survey Data with Embeddings (Overview)

- Hands on projects, and switch to Data science career path from research
- Get to know Shaw better! Plus, have a clear path with resources to guide you on where to go and what to do for the future
- Fundamentals understanding, Hands on skills, small projects running in diff frameworks, create a small GH portfolio.
- Build multiple AI applications
- I would like to learn how to practically setup LLM application in a production environment so that I can start creating some AI web apps for internal and external use.
- Ability to train model based on a data set, and how to do predictive analysis.
- Develop a good foundation in AI/ML.
- I love to figure out good strategies to use in my software solutions using AI
- Learn to decide and advice between different data science archictures and options
- Implement practical AI cases
- Foundation in AI/ML
- Land a program manager job working with AI product/program
- Build an app
- able to create a LLM from scratch
- Get a new job
- At my former job I had a coworker who had the tedious task of making a weekly report summarizing local news for the boss. At the time I was completely sure that was something AI could do, but I didn't have the time nor the knowledge to develop such a thing. I constantly find myself having ideas like this, so a dream outcome would be finishing the course with at least a roadmap for making such an app.
- Build cool products
- Be able to be comfortable with GenAi
- A certificate to showcase my new skills, A jump start to do my own programing and be able to communicate with programmers
- Setup my own environment to compare Machine Learning statistics created by my companies Data Scientists against my own environment.
- Be able to create and deploy my own Ai powered apps
- Build and launch a product
- Getting skills to build AI technologies for many projects
- Being able to implement llms into projects
- Landing a new job
- to deepen my understanding of advanced AI techniques.And I also aim to expand my professional network and collaborate with like-minded individuals to explore new opportunities in the AI landscape.
- Be able to train a model and have some practical usage of llm.
- Learn to identify the ML solution and lead projects based on AI
- Able to put into practice AI/ML for real bunsiess solutions
- Master AI powered productivity tools to streamline regulatory compliance work
- Learn Python related to AI. Implement prototypes.
- 1) build GPT that accesses functions and APIs, 2) hands-on fine-tune a model using LoRA 3) write and debug Python code (with AI assistance) that accesses ChatGPT, Perplexity and Google search, 4) build working agent(s) that can output results in less than 10 seconds 5) possibly use a RAG efficiently for booklength PDF texts
- Fully understand AI and LLM, and how to build one using Python
- Be able to comfortably add AI into daily or business use cases
- Build my own AI product with the AI services available these days
- I want to build my own MicroSaaS products
- I love to apply ai solution to existing and future back-end projects



Example 1

Analyzing Survey Data with Embeddings (Flowchart)

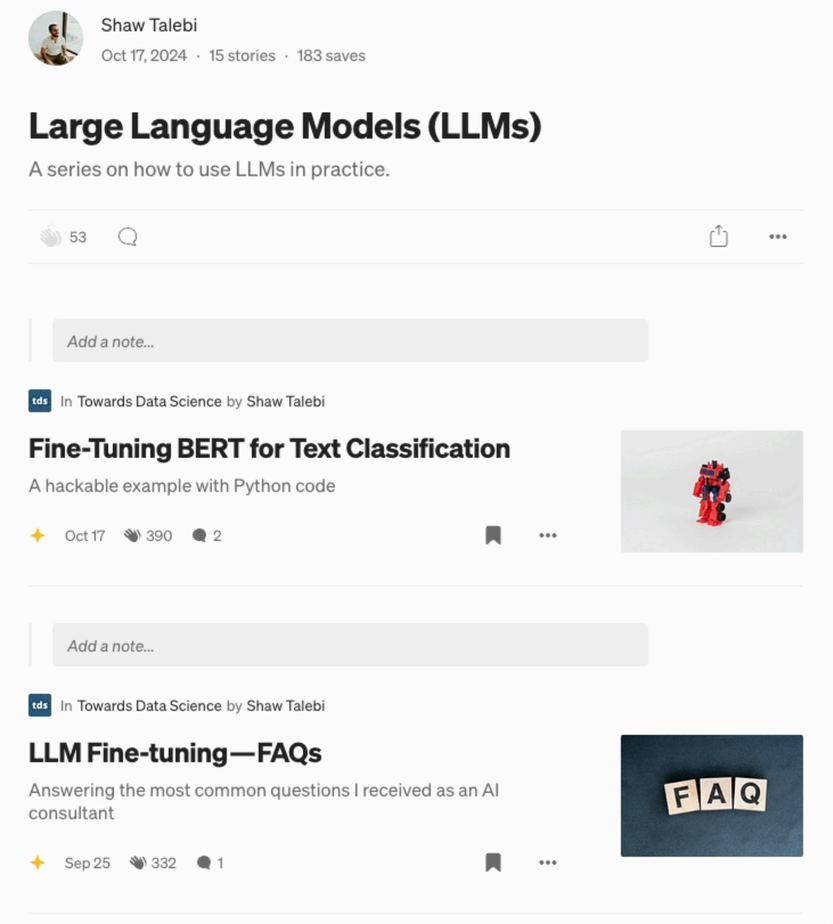


Example 1

Analyzing Survey Data with Embeddings (Code)

Example 2

Semantic Search + RAG with LlamaIndex (Motivation)

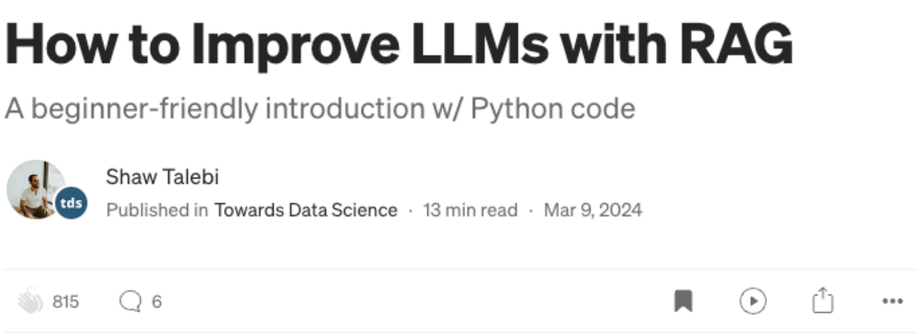


Finding specific information across several resources can be tedious

Example 2

Semantic Search + RAG with LlamaIndex (Overview)

What is RAG?



1. **Article title:** How to Improve LLMs with RAG
   Section: What is RAG?
   Snippet: RAG works by adding a step to this basic process. Namely, a retrieval step is performed where, based on the user's prompt, the relevant information is extracted from an external knowledge base and injected into the prompt before being passed to the LLM.
2. **Article title:** How to Improve LLMs with RAG
   Section: What is RAG?
   Snippet: The basic usage of an LLM consists of giving it a prompt and getting back a response.
3. **Article title:** How to Improve LLMs with RAG
   Section: How it works
   Snippet: There are 2 key elements of a RAG system: a retriever and a knowledge base.

Requires digging to answer question

Example 2

Semantic Search + RAG with LlamaIndex (Overview)

What is RAG?

1. **Article title:** How to Improve LLMs with RAG
   Section: What is RAG?
   Snippet: RAG works by adding a step to this basic process. Namely, a retrieval step is performed where, based on the user's prompt, the relevant information is extracted from an external knowledge base and injected into the prompt before being passed to the LLM.

2. **Article title:** How to Improve LLMs with RAG
   Section: What is RAG?
   Snippet: The basic usage of an LLM consists of giving it a prompt and getting back a response.

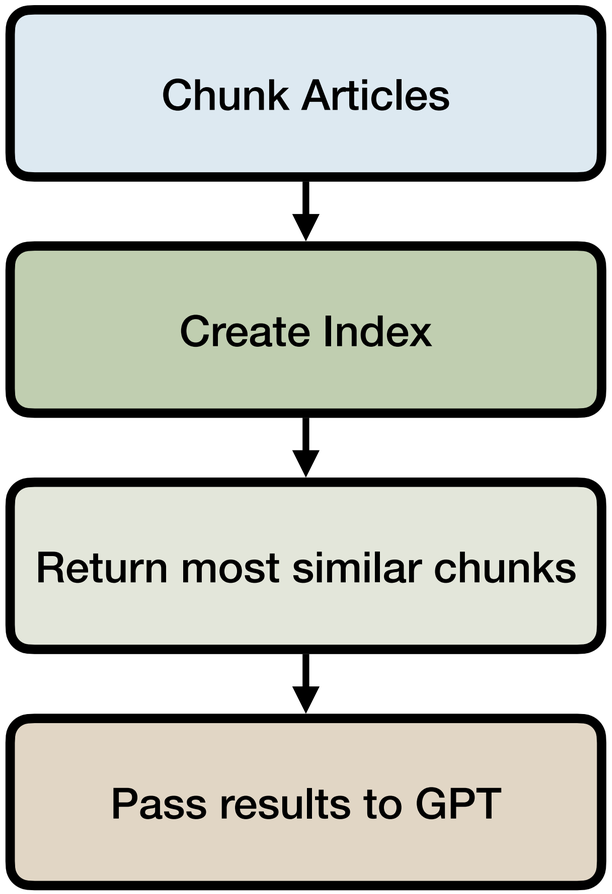
3. **Article title:** How to Improve LLMs with RAG
   Section: How it works
   Snippet: There are 2 key elements of a RAG system: a retriever and a knowledge base.

RAG, or Retrieval Augmented Generation, is an approach that enhances the functionality of large language models (LLMs) by incorporating a retrieval step. This process involves extracting relevant information from an external knowledge base based on the user's prompt and injecting it into the prompt before passing it to the LLM. Essentially, while the basic operation of an LLM remains the same—receiving a prompt and generating a response—RAG augments this process by utilizing additional information to improve the quality and relevance of the output (Snippets 1, 2, 9).

A RAG system consists of two key components: a retriever, which fetches relevant information, and a knowledge base, which stores the information that can be accessed by the LLM (Snippets 3, 7). The effectiveness of a RAG system is influenced by how well the source documents are prepared, as poorly formatted documents can complicate the extraction of useful information (Snippet 5).

Example 2

Semantic Search + RAG with LlamaIndex (Flowchart)

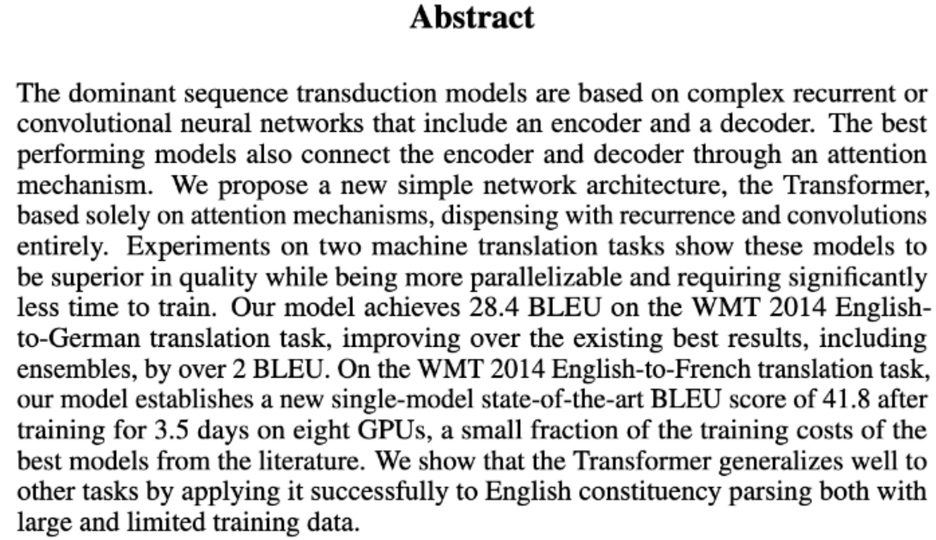


Example 2

Semantic Search + RAG with LlamaIndex (Code)

Example 3

PDF Parsing with Docling (Motivation)



Abstract

The dominant sequence transduction models are based on complex recurrent or convolutional neural networks that include an encoder and a decoder. The best performing models also connect the encoder and decoder through an attention mechanism. We propose a new simple network architecture, the Transformer, based solely on attention mechanisms, dispensing with recurrence and convolutions entirely. Experiments on two machine translation tasks show these models to be superior in quality while being more parallelizable and requiring significantly less time to train. Our model achieves 28.4 BLEU on the WMT 2014 English-to-German translation task, improving over the existing best results, including ensembles, by over 2 BLEU. On the WMT 2014 English-to-French translation task, our model establishes a new single-model state-of-the-art BLEU score of 41.8 after training for 3.5 days on eight GPUs, a small fraction of the training costs of the best models from the literature. We show that the Transformer generalizes well to other tasks by applying it successfully to English constituency parsing both with large and limited training data.

PyMuPDF

```
Abstract
The dominant sequence transduction models are based on complex recurrent or
convolutional neural networks that include an encoder and a decoder. The best
performing models also connect the encoder and decoder through an attention
mechanism. We propose a new simple network architecture, the Transformer,
based solely on attention mechanisms, dispensing with recurrence and convolutions
entirely. Experiments on two machine translation tasks show these models to
be superior in quality while being more parallelizable and requiring significantly
less time to train. Our model achieves 28.4 BLEU on the WMT 2014 English-
to-German translation task, improving over the existing best results, including
ensembles, by over 2 BLEU. On the WMT 2014 English-to-French translation task,
our model establishes a new single-model state-of-the-art BLEU score of 41.8 after
training for 3.5 days on eight GPUs, a small fraction of the training costs of the
best models from the literature. We show that the Transformer generalizes well to
other tasks by applying it successfully to English constituency parsing both with
large and limited training data.
```

Example 3

PDF Parsing with Docling (Motivation)

Table 1: Maximum path lengths, per-layer complexity and minimum number of sequential operations for different layer types. n is the sequence length, d is the representation dimension, k is the kernel size of convolutions and r the size of the neighborhood in restricted self-attention.

Layer Type	Complexity per Layer	Sequential Operations	Maximum Path Length
Self-Attention	$O(n^2 \cdot d)$	$O(1)$	$O(1)$
Recurrent	$O(n \cdot d^2)$	$O(n)$	$O(n)$
Convolutional	$O(k \cdot n \cdot d^2)$	$O(1)$	$O(log_k(n))$
Self-Attention (restricted)	$O(r \cdot n \cdot d)$	$O(1)$	$O(n/r)$

PyMuPDF

```
Table 1: Maximum path lengths, per-layer complexity and minimum number of sequential operations
for different layer types. n is the sequence length, d is the representation dimension, k is the kernel
size of convolutions and r the size of the neighborhood in restricted self-attention.
Layer Type
Complexity per Layer
Sequential
Maximum Path Length
Operations
Self-Attention
O(n2 · d)
O(1)
O(1)
Recurrent
O(n · d2)
O(n)
O(n)
Convolutional
O(k · n · d2)
O(1)
O(logk(n))
Self-Attention (restricted)
O(r · n · d)
O(1)
O(n/r)
3.5
```

Example 3

PDF Parsing with Docling (Motivation)

Table 1: Maximum path lengths, per-layer complexity and minimum number of sequential operations for different layer types. n is the sequence length, d is the representation dimension, k is the kernel size of convolutions and r the size of the neighborhood in restricted self-attention.

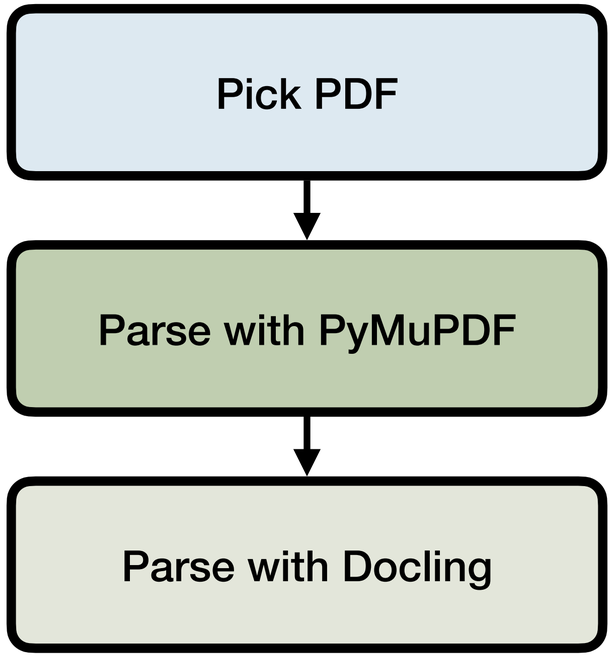
Layer Type	Complexity per Layer	Sequential Operations	Maximum Path Length
Self-Attention	$O(n^2 \cdot d)$	$O(1)$	$O(1)$
Recurrent	$O(n \cdot d^2)$	$O(n)$	$O(n)$
Convolutional	$O(k \cdot n \cdot d^2)$	$O(1)$	$O(log_k(n))$
Self-Attention (restricted)	$O(r \cdot n \cdot d)$	$O(1)$	$O(n/r)$

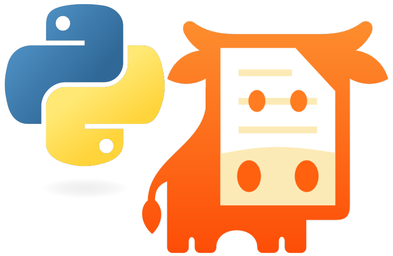
Docling →

Layer Type	Complexity per Layer	Sequential Operations	Maximum Path Length
Self-Attention	O (n 2 · d)	O (1)	O (1)
Recurrent	O (n · d 2)	O (n)	O (n)
Convolutional	O (k · n · d 2)	O (1)	O (log k (n))
Self-Attention (restricted)	O (r · n · d)	O (1)	O (n/r)

Example 3

PDF Parsing with Docling (Flowchart)





Example 3

PDF Parsing with Docling (Code)

Homework 4

Project 💻

Solve a Problem with Text Embeddings

Pre-work ✍️

Session 4: AI Agents

References

[1] Multimodal RAG: Process Any File Type with AI

[2] How to Improve LLMs with RAG

[3] Text Embeddings, Classification, and Semantic Search

[4] Text Embeddings, Classification, and Semantic Search (w/ Python Code)

Similarity Scores

Scaling cosine similarity with softmax

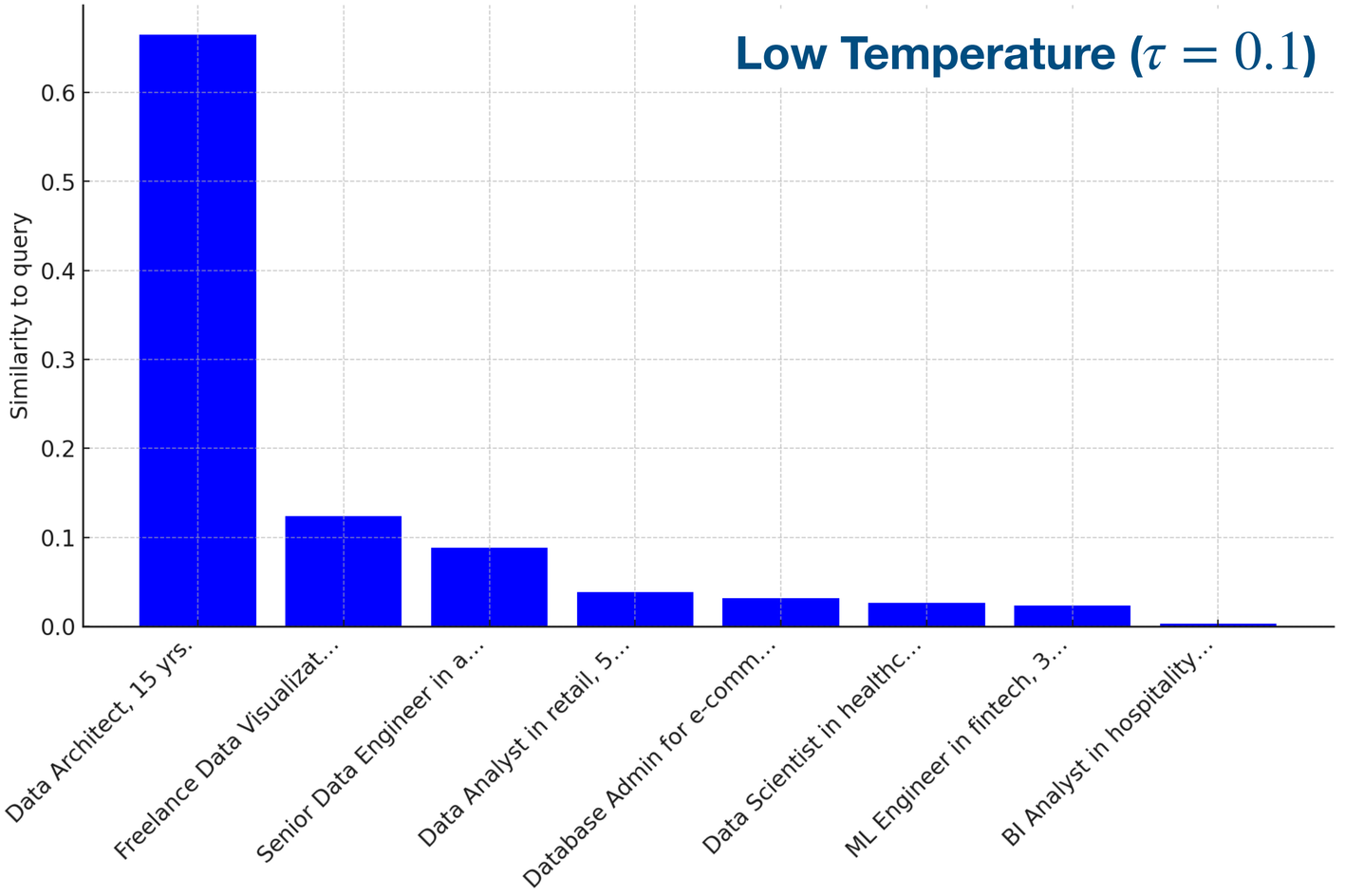
Softmax Function

$$\bar{s}_i = \frac{e^{\frac{s_i}{\tau}}}{\sum e^{\frac{s_i}{\tau}}}$$

s_i = cosine similarity between $\vec{q}$ and $\vec{d}_i$

τ = Temperature $\quad \sum \bar{s}_i = 1$

Probability-like score!



Similarity Scores

Scaling cosine similarity with softmax

Softmax Function

$$\bar{s}_i = \frac{e^{\frac{s_i}{\tau}}}{\sum e^{\frac{s_i}{\tau}}}$$

s_i = cosine similarity between $\vec{q}$ and $\vec{d}_i$

τ = Temperature $\quad \sum \bar{s}_i = 1$

Probability-like score!