Design + Code-Walkthrough

Search-engine for Q&A

AppliedAICourse.com

APPLIED COURSE

Agenda:

1. Pre-req
2. Problem definition & solution requirement
3. Dataset [+code]
4. High level design & Data-structures
5. Docker containirazation [+setup]

6. Sentence-vectors [code]

7. Elastic Search [installation]

8. Index data [code]

9. Search [code]

10. Deployment [code]

11. Extensions & optional assignment

References:

Blog: https://www.elastic.co/blog/text-similarity-search-with-vectors-in-elasticsearch

Code: https://github.com/jtibshirani/text-embeddings

Terms related to ElasticSearch:
https://www.elastic.co/guide/en/elasticsearch/reference/current/glossary.html

Simple tutorial on Python+ES: https://medium.com/naukri-engineering/elasticsearch-tutorial-for-beginners-using-python-b9cb48edcedc

Pre-req:

1. Python programming
2. Inbuilt DS in python : lists, dict, tuples, sets

We will NOT be able to dive deep into all the concepts

Problem - def:

→ Q^i

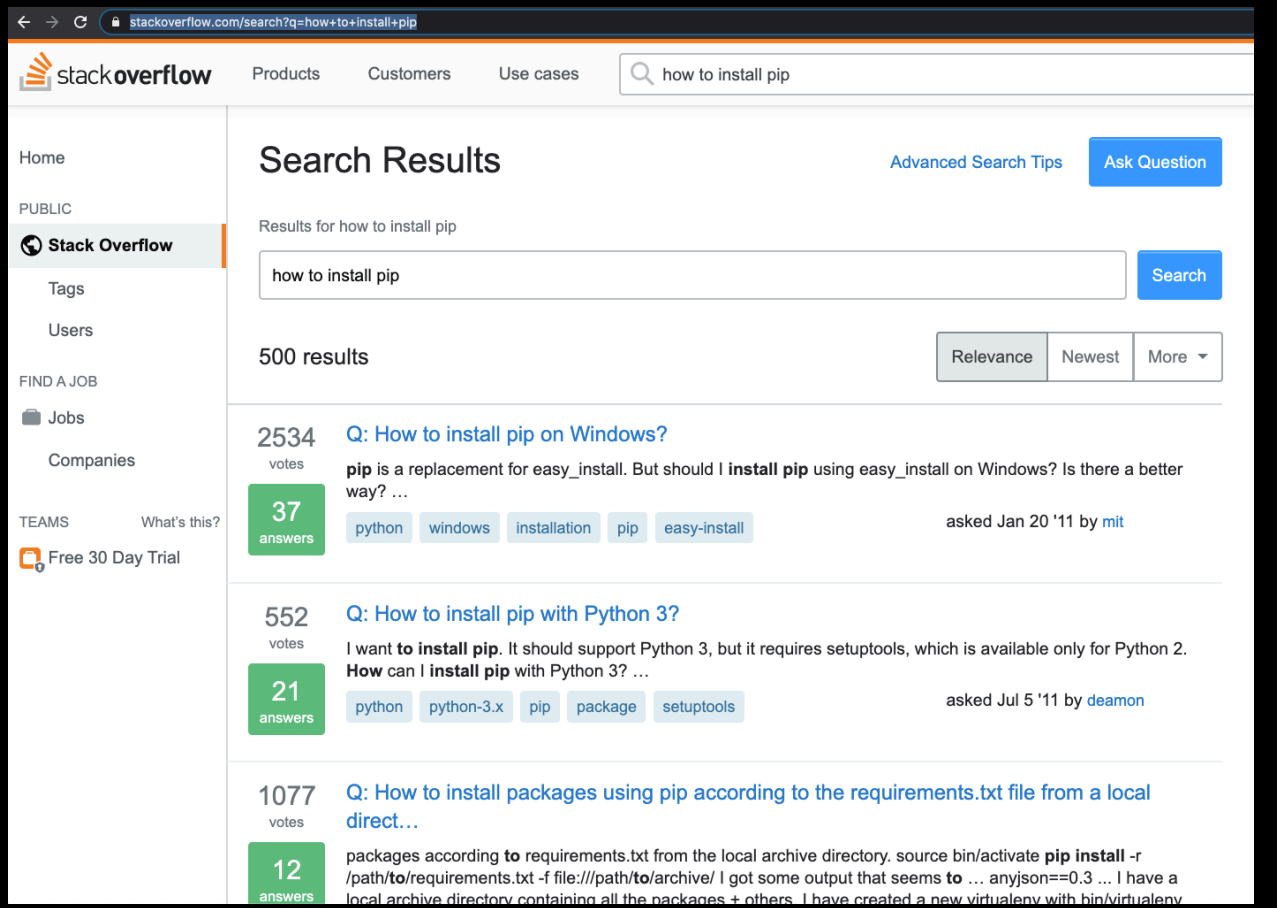
A^i_1

A^i_2

⋮

A^i_K

→ fast (< 500 ms)



→ high precision & recall

→ low computational / server costs [QPS: #servers: cost per server:]

→ quick to build & deploy [< 7 days] a basic version

Dataset

STACK SAMPLE: https://www.kaggle.com/stackoverflow/stacksample → fields/columns

→ DOWNLOAD

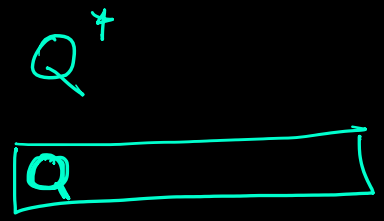
→ UNZIP xxxxx.zip

→ count # questions & answers [readData.py]

HLD + Data Structures

Q^i : id, title, body, ...

A^j : id, body, parentId, ...



design choices : Q^i title → simplest

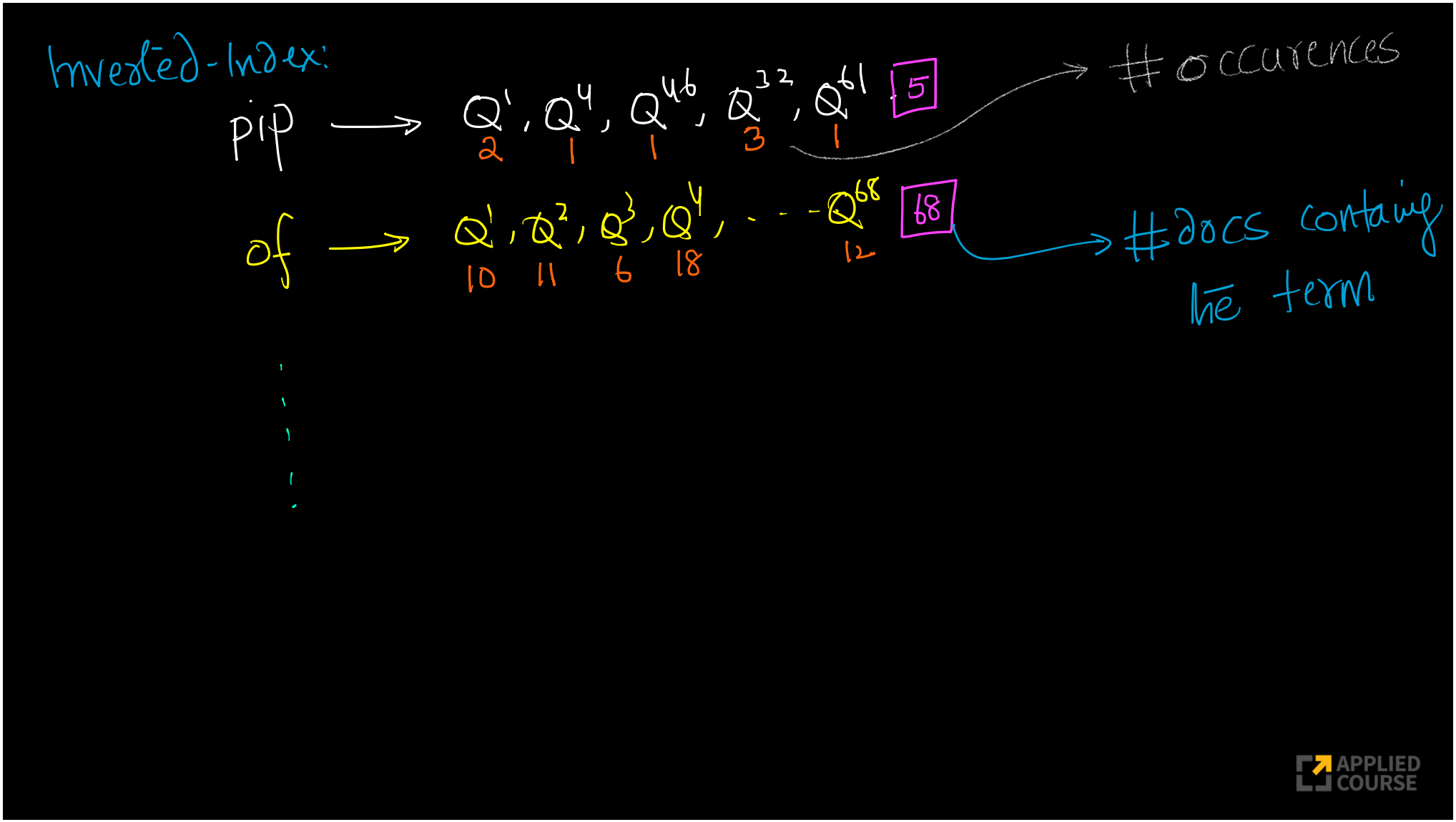
+ Q^i body

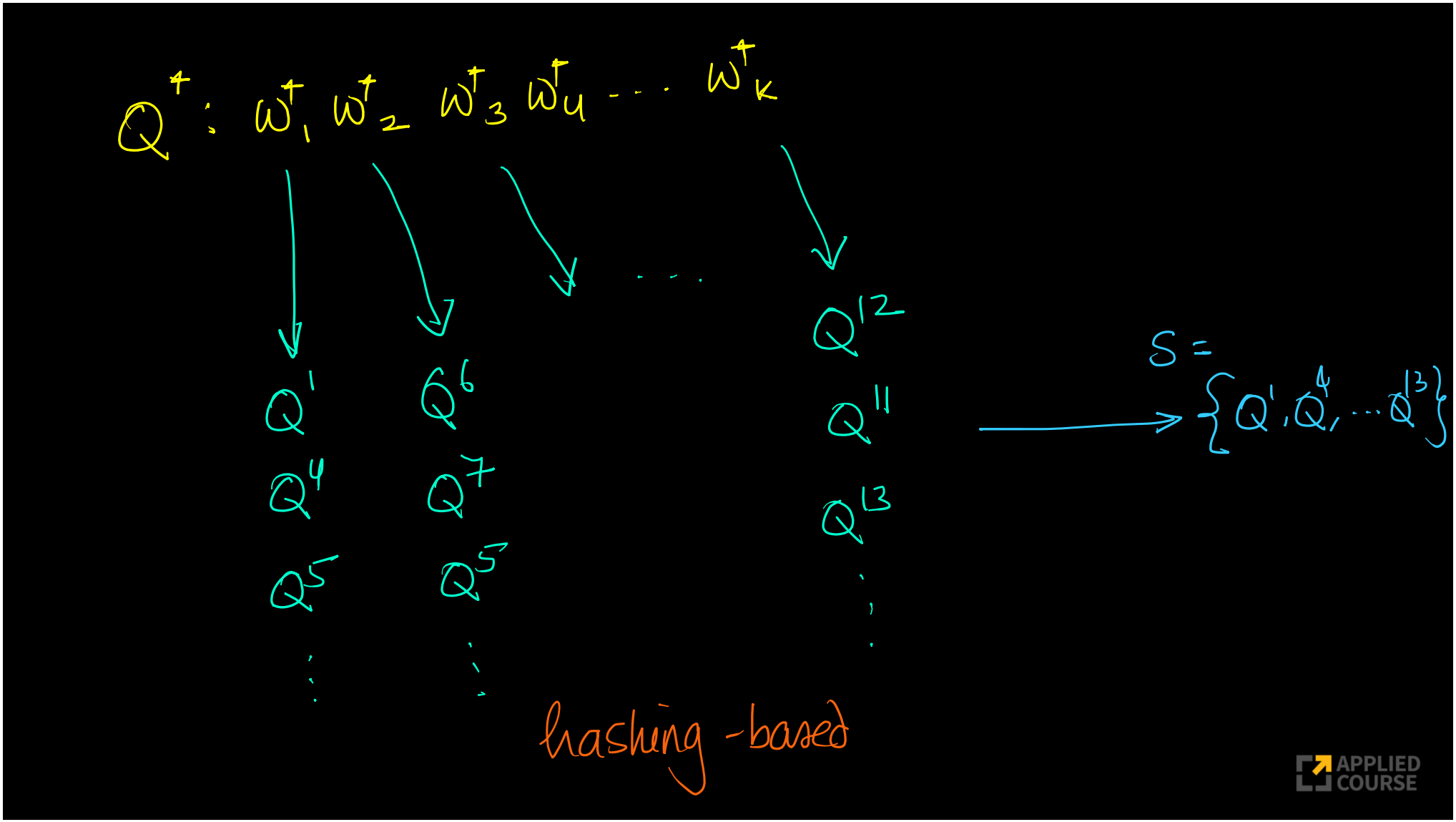
+ A^j body

Q^i : w_1^i w_2^i w_3^i ---- w_d^i

title

Search $\Rightarrow$ hashing [O(1)]


Inverted-Index:
pip → Q1, Q4, Q46, Q32, Q61
2 1 1 3 1
5
occurences
of → Q1, Q2, Q3, Q4, ... Q68
10 11 6 18 12
68
docs containing the term


Q^t : w_1^t w_2^t w_3^t w_4^t ... w_k^t
Q^1
Q^4
Q^5
Q^6
Q^7
Q^5
Q^{12}
Q^{11}
Q^{13}
S = $\{Q^1, Q^4, \ldots Q^{13}\}$
hashing-based

score (Q^*, Q^i) $\forall Q^i \in S$

TF-IDF based

Term-freq

occurences

Inverse doc-freq

eg: of, the, if, of ...

pip, install, ...

Elastic Search : Inverted Index, scoring, distributed, realtime, ...