Search and Recommendation

Hao Sheng

August 10th, 2023

ICME

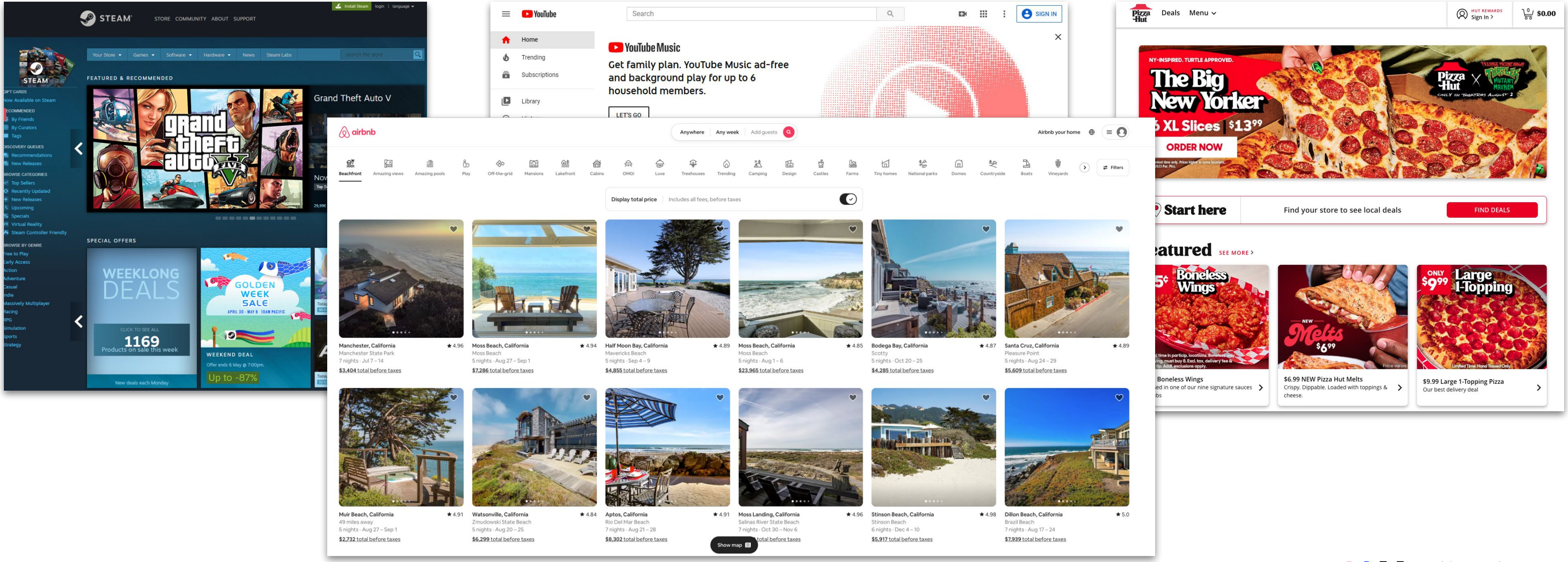
Agenda

- **Recap**
- **Lecturing:** Search Engine System
- **Break-out:** Measure the Success of the Recommendation System
- **Lecturing**: Advanced Topics (Part I)
- **Lab:** Recommendation system notebook II
- **Lecturing**: Advanced Topics (Part II)

60% Lecturing | 25% Lab | 10% Discussion

Recap of Day 1

Homepages @ 2023

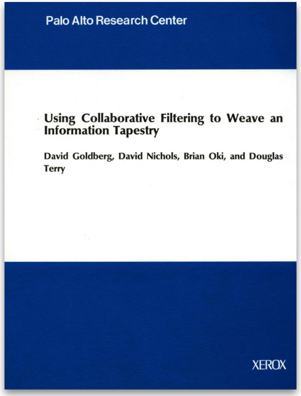


Mentimeter

Everyone gives user recommendations on the first impression!

Earlier Recommendation Systems

- Recommendation system: RS automatically selects personalized information based on users' preferences.
- Grundy:
  - Ask user questions and assign stereotype.
  - **Content-based filtering.**
- Tapestry:
  - Find similar users and recommend their choices.
  - **Collaborative filtering.**




COGNITIVE SCIENCE 3, 329-354 (1979)

User Modeling via Stereotypes *

ELAINE RICH

The University of Texas at Austin

This paper addresses the problems that must be considered if computers are going to treat their users as individuals with distinct personalities, goals, and so forth. It first outlines the issues, and then proposes stereotypes as a useful mechanism for building models of individual users on the basis of a small amount of information about them. In order to build user models quickly, a large amount of uncertain knowledge must be incorporated into the models. The issue of how to resolve the conflicts that will arise among such inferences is discussed. A system, Grundy, is described that bunds models of its users, with the aid of stereotypes, and then exploits those models to guide it in its task, suggesting novels that people may find interesting. If stereotypes are to be useful to Grundy, they must accurately characterize the users of the system. Some techniques to modify stereotypes on the basis of experience are discussed. An analysis of Grundy's performance shows that its user models are effective in guiding its performance.

1. INTRODUCTION

Scene I

Someone walks into a large library, tells the librarian that he is interested in China, and asks for some books. What sort of books does the librarian recommend? That depends. Is the person a small child who just saw a TV show about China and wants to see more pictures of such an exotic place? Is the person a high school student doing a term paper? Or maybe a prospective tourist? Or a scholar interested in Eastern thought? Can the person read Chinese? The librarian needs to know these things before he can point the reader to the right books. Some of what he needs to know he'll know before he even thinks about it, such as the approximate age of the person. Some things he'll assume until he has evidence to the contrary, such as that the person does not read Chinese. To find out other things, he'll ask a few specific questions. Only after he has a rough model of the person he's talking to can he answer the question.

Scene II

The phone rings in the information division of a large pharmaceutical firm. The caller wants information about a drug the company makes. What sort of information should be provided? That depends. Is the caller a doctor, a patient, or an FDA representative? To provide the right information, the person answering the phone needs to know some facts about the caller.

The scenes above illustrate some kinds of situations in which people need to form a *model* of the person with whom they are dealing before they can behave appropriately. They form their model by collecting a few specific pieces of information and then invoking the knowledge they have about the groups to which the current person belongs, such as scholar or medical patient.

As computers come to be used by a larger number of people to help perform a great variety of tasks, it is becomming more and more important for them to be easy for people to use. There are many factors than can contribute to the ease of use of a computer system, ranging from the good design of input devices such as terminals to the speed of the system's response, the appropriateness of its response, and the naturalness of its input and output languages. Appropriate models of the users of a system can be an important contribution because they can simultaneously affect several of those factors, such as speed and quality of response and habitability of the language interface.

Most systems that interact with human users contain, even if only implicitly, some sort of model of the creatures they will be dealing with. For example, the central assumption behind the mini-max strategy used by game playing programs is that the opponent is trying to win and will therefore make his best possible move. Although it is almost always valid to assume that the opponent wants to win, it is much less often true that he will therefore make the best move. He may, and probably does, have idiosyncracies of style or strategy that preclude that. Of course, human players know that and watch for evidence of such quirks in their opponents.
The term "user model" can be used to mean several different things. The three major dimensions along which user models can be classified are:

Quick Recap: Collaborative Filtering

Sara	5	3		2	2	2
Jesper	4	3	4		3	3
Therese	5	2	5	2	1	1
Helle	3	5	3		1	1
Pietro	3	3	3	2	4	5
Ekaterina	2	3	2	3	5	5




Evaluate Recommendation Systems: Online Evaluation

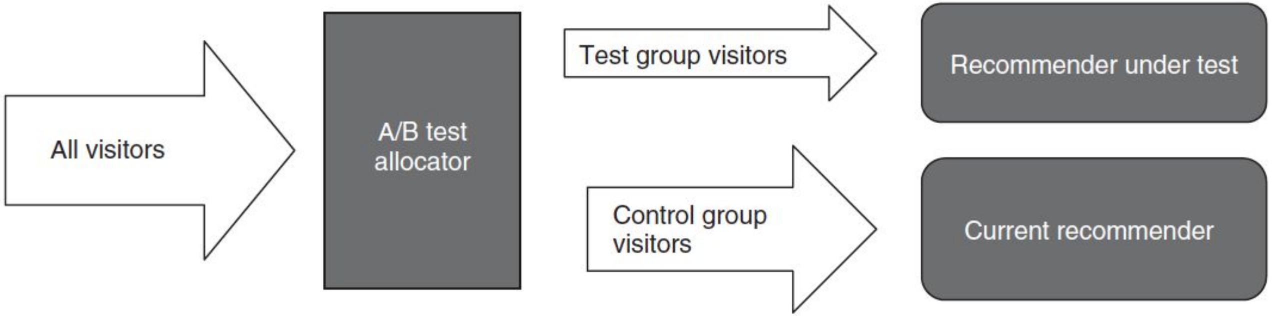
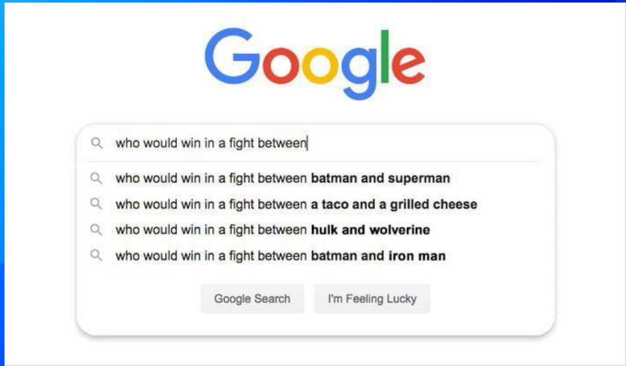


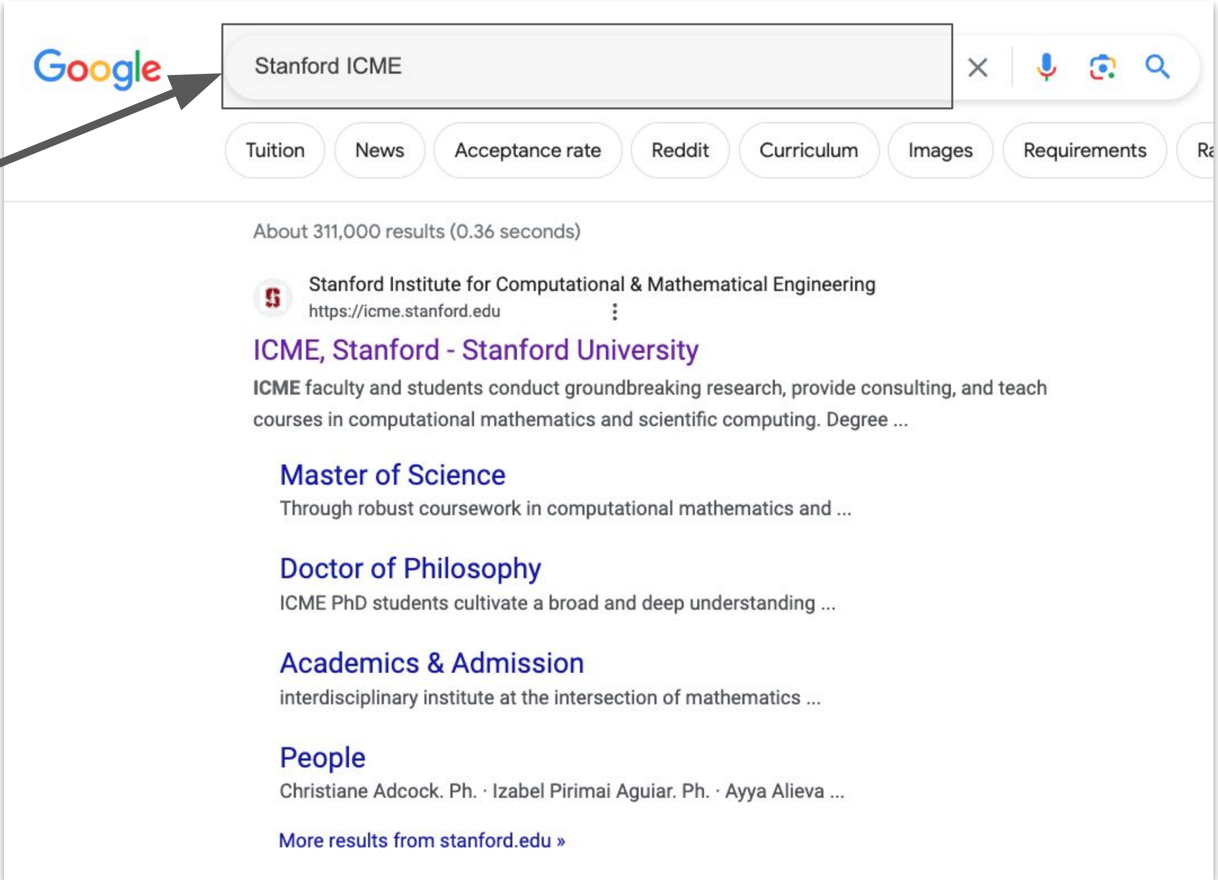
Figure 9.16 In an A/B test, visitors are split into two groups: the test group that sees the new feature and a control group that continues as usual.

Search Engine System



Search Engine: First Glance

Search Query



Search Engine: First Glance

Stanford University
https://events.stanford.edu › department › institute_for...

Institute for Computational and Mathematical Engineering ...

The Institute for Computational & Mathematical Engineering (**ICME**) is a degree granting (M.S./Ph.D.) interdisciplinary institute at the intersection of ...

Stanford University Bulletin
https://bulletin.stanford.edu › programs › CME-MS

CME-MS Program - Stanford Bulletin

ICME is a degree granting (M.S./Ph.D.) interdisciplinary institute at the intersection of mathematics, computing, engineering and applied sciences.

Courseor course: Intermediate Econometrics II (3 ...

Stanford Bulletin Archive
https://archived-bulletin.stanford.mobi › instituteforco...

Institute for Computational and Mathematical Engineering

At **ICME**, we design state-of-the-art mathematical and computational models, methods, and algorithms for engineering and science applications. The program ...

Or MS&E 327: Topics in Causal Inference   STATS 263: Design of Experiments

LinkedIn
https://www.linkedin.com › company › icme-stanford

Institute for Computational and Mathematical Engineering ...

Institute for Computational and Mathematical Engineering at **Stanford** University (**ICME**) | 1056 followers on LinkedIn. Groundbreaking research into complex ...

Search Engine Results
(Websites)

Search Engine: First Glance

Stanford Online
https://online.stanford.edu › programs › computationa...

Computational and Mathematical Engineering MS Degree

The Institute for Computational and Mathematical Engineering (**ICME**) is a degree granting institute at the intersection of mathematics, computing, engineering ...

https://twitter.com/ICMEStanford

Stanford ICME (@ICMEStanford) · Twitter

Stanford ICME's new Generative Models workshop—from 8/9 to 8/10 & taught by instructor Aashwin Mishra—focuses on models that generate new data instances, such as images, text, or audio.

Register: www.eventbrite.com/e/ic...

Twitter · Jul 28, 2023

Explore methods for enhancing #UX & retrieving information in @Stanford ICME's new Search and Recommendation workshop from 8/9 to 8/10, led by @Apple Staff Machine Learning Engineer @hao_ssr.

Register: www.eventbrite.com/e/ic...

Twitter · Jul 27, 2023

Learn the building blocks of modern #NLP concepts in @Stanford ICME's Intro course from 8/7 to 8/8, taught by brothers and @Google Software Engineers @afshinea & @shervinea.

Register: www.eventbrite.com/e/ic...

Twitter · Jul 26, 2023

Search Engine Results (Tweets)

Stanford University
https://events.stanford.edu › department › institute_for...

Institute for Computational and Mathematical Engineering ...

The Institute for Computational & Mathematical Engineering (**ICME**) is a degree granting (M.S./Ph.D.) interdisciplinary institute at the intersection of ...

Search Engine: First Glance

People also ask

How competitive is Stanford graduate school?

What is the Toefl code for Stanford ICME?

Is Stanford University good for engineering?

What is the mathematical and computational finance program at Stanford University?

Feedback

Stanford Online
https://online.stanford.edu › programs › computationa...

Computational and Mathematical Engineering MS Degree

The Institute for Computational and Mathematical Engineering (**ICME**) is a degree granting institute at the intersection of mathematics, computing, engineering ...

Search Engine Results
(Tweets)

Search Engine v.s. Movie Recommendation

- It has a search bar!
- The items are mal-defined at the first glance.
- User does not simply rate the search results!

Search Engine as a Recommendation System

"Recommender systems (RSs) are software tools and techniques that provide suggestions for items that are most likely of interest to a particular user. "

--- *Introduction to Recommender Systems Handbook*

Search Engine as a Recommendation System

“Recommender systems (RSs) are software tools and techniques that provide suggestions for items that are most likely of interest to a particular user. ”

--- *Introduction to Recommender Systems Handbook*

Search Engine is a **Recommendation System!**

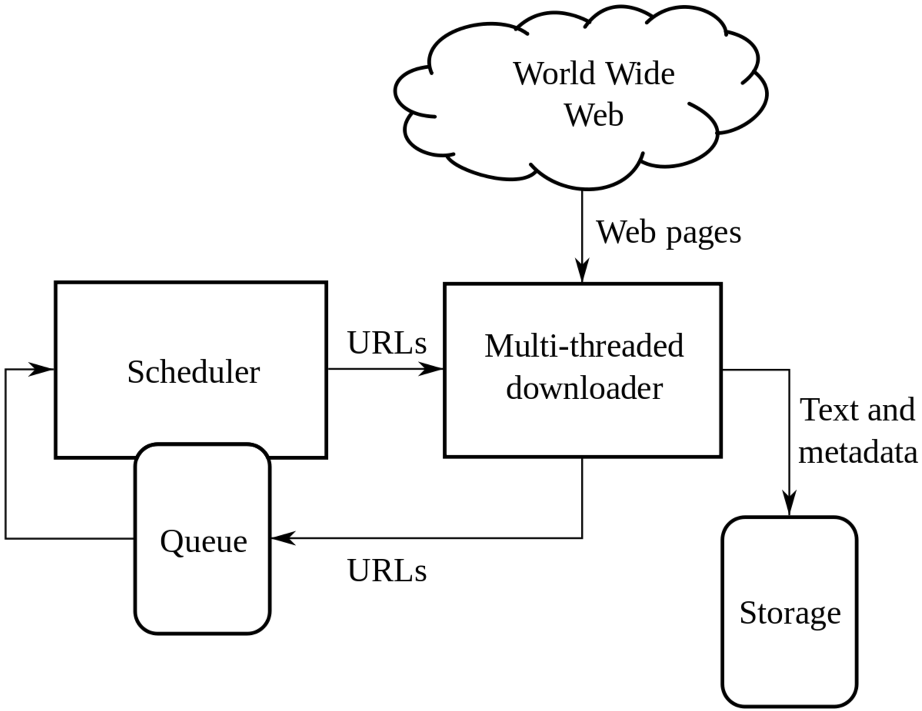
Search Engine: Learning Goals

- It has a search bar!
  - -> How to incorporate the user intention?
- The items are mal-defined at the first glance.
  - -> How to crawl the internet and store the items?
- User does not simply rate the search results!
  - -> How to assign user-item rating with user data?

Search Engine: Web Crawler

Where to get the search results (items)?

The internet!



Search Engine: Web Crawler - Recursion

Where to get the search results (items)?

The internet!

```
class Spider:
    name = 'icme_spider'
    start_urls = 'https://icme.stanford.edu/'
    parsed_urls = []

    def parse(self, url: str):
        self.parse_url.append(url)
        for next_url in Website(url):
            self.parse(next_url)
```

Search Engine: Web Crawler - Recursion

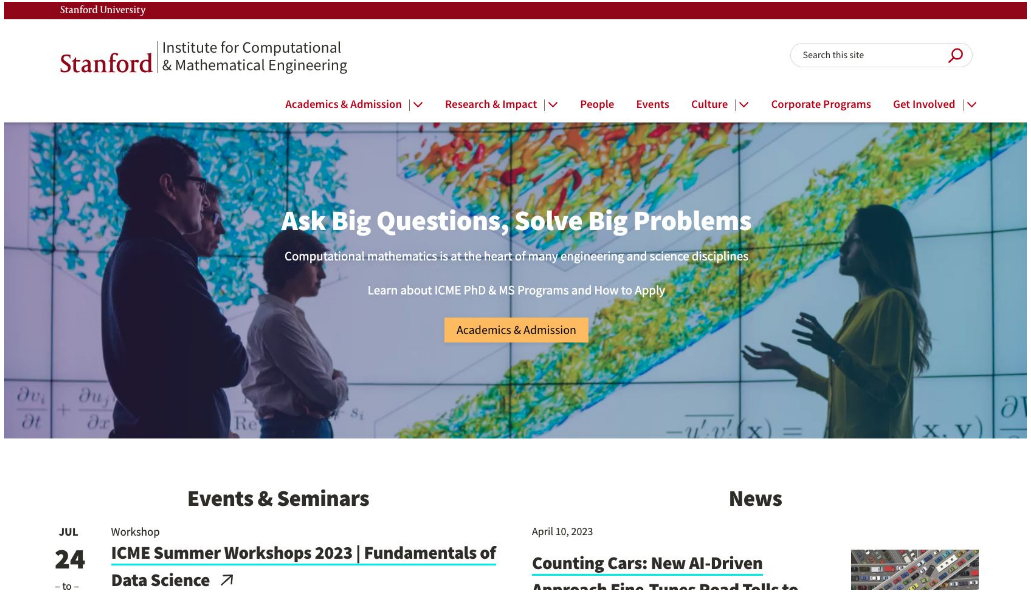
parsed_urls = [“https://icme.stanford.edu”]

```
class Spider:
    name = 'icme_spider'
    start_urls = 'https://icme.stanford.edu/'
    parsed_urls = []

    def parse(self, url: str):
        self.parse_url.append(url)
        for next_url in Website(url):
            self.parse(next_url)
```

Search Engine: Web Crawler - Recursion

parsed_urls = [“https://icme.stanford.edu”]

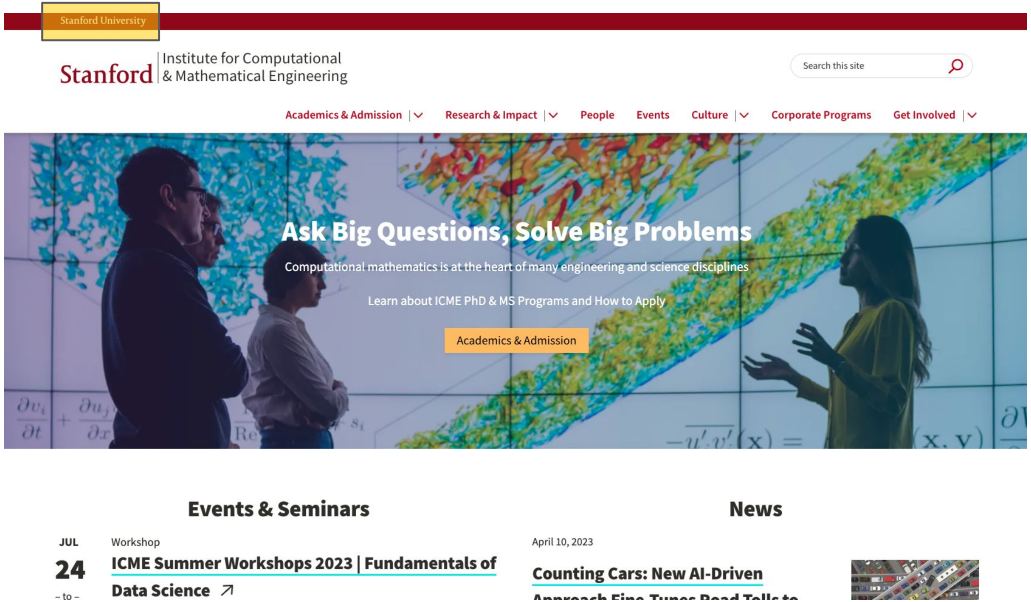
```
class Spider:
    name = 'icme_spider'
    start_urls = 'https://icme.stanford.edu/'
    parsed_urls = []

    def parse(self, url: str):
        self.parse_url.append(url)
        for next_url in Website(url):
            self.parse(next_url)
```



Search Engine: Web Crawler - Recursion

parsed_urls = [“https://icme.stanford.edu”]

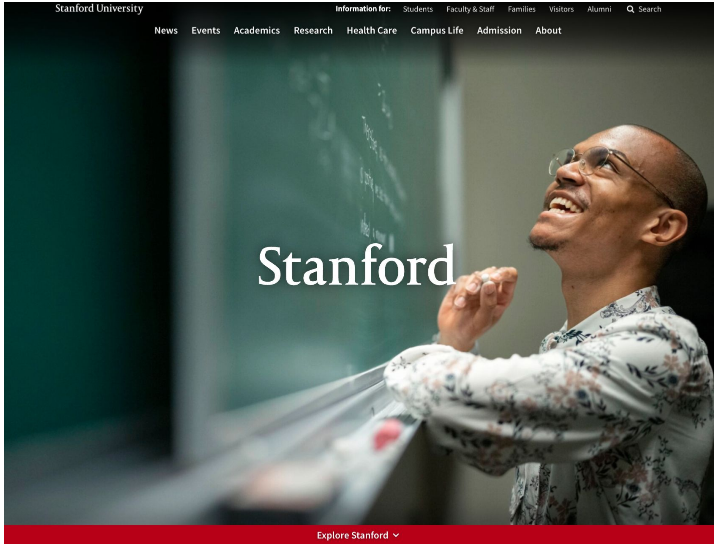
```
class Spider:
    name = 'icme_spider'
    start_urls = 'https://icme.stanford.edu/'
    parsed_urls = []

    def parse(self, url: str):
        self.parse_url.append(url)
        for next_url in Website(url):
            self.parse(next_url)
```



Search Engine: Web Crawler - Recursion

parsed_urls = [“https://icme.stanford.edu”]

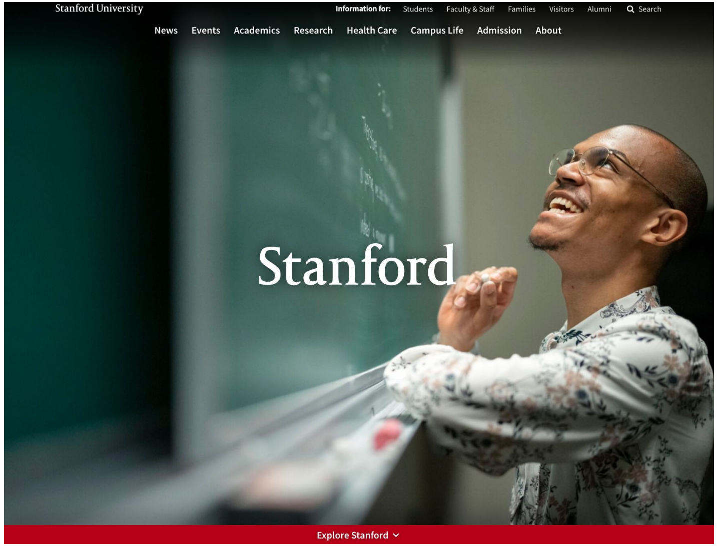
```python
class Spider:
    name = 'icme_spider'
    start_urls = 'https://icme.stanford.edu/'
    parsed_urls = []

    def parse(self, url: str):
        self.parse_url.append(url)
        for next_url in Website(url):
            self.parse(next_url)
```



Search Engine: Web Crawler - Recursion

```python
class Spider:
    name = 'icme_spider'
    start_urls = 'https://icme.stanford.edu/'
    parsed_urls = []

    def parse(self, url: str):
        self.parse_url.append(url)
        for next_url in Website(url):
            self.parse(next_url)
```

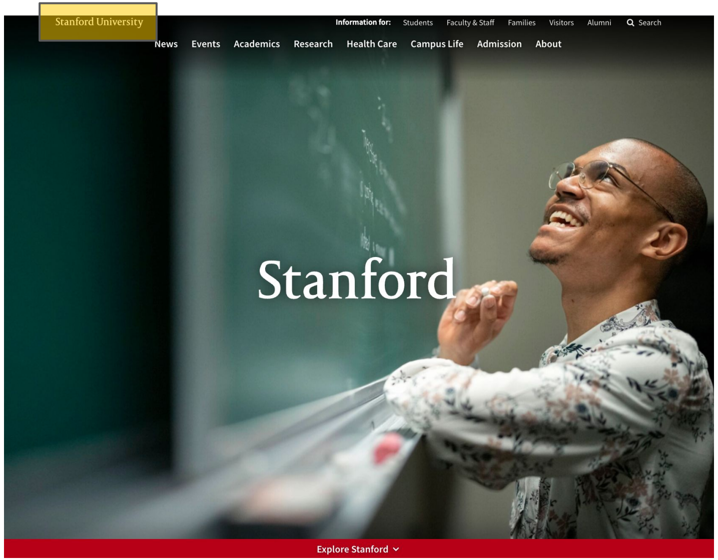
parsed_urls = [“https://icme.stanford.edu”,
“https://www.stanford.edu/”
]



Search Engine: Web Crawler - Recursion

```
class Spider:
    name = 'icme_spider'
    start_urls = 'https://icme.stanford.edu/'
    parsed_urls = []

    def parse(self, url: str):
        self.parse_url.append(url)
        for next_url in Website(url):
            self.parse(next_url)
```

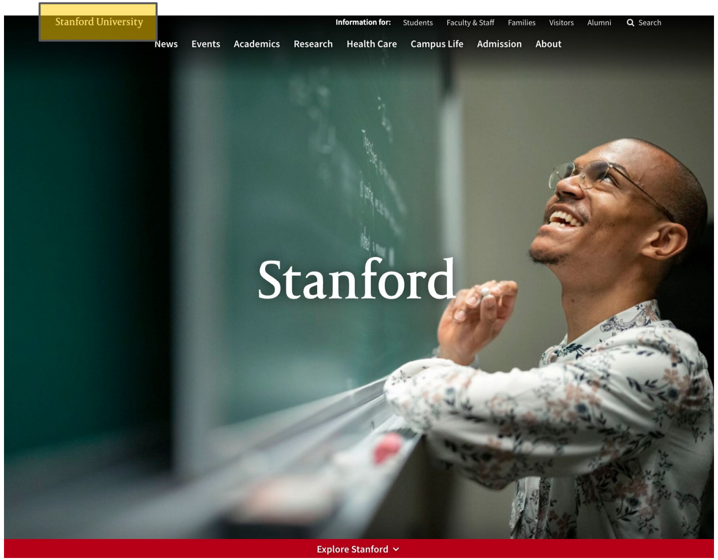
parsed_urls = ["https://icme.stanford.edu", "https://www.stanford.edu/"
]



Search Engine: Web Crawler - Recursion

```python
class Spider:
    name = 'icme_spider'
    start_urls = 'https://icme.stanford.edu/'
    parsed_urls = []

    def parse(self, url: str):
        self.parse_url.append(url)
        for next_url in Website(url):
            self.parse(next_url)
```

parsed_urls = [“https://icme.stanford.edu”,
“https://www.stanford.edu/”
]

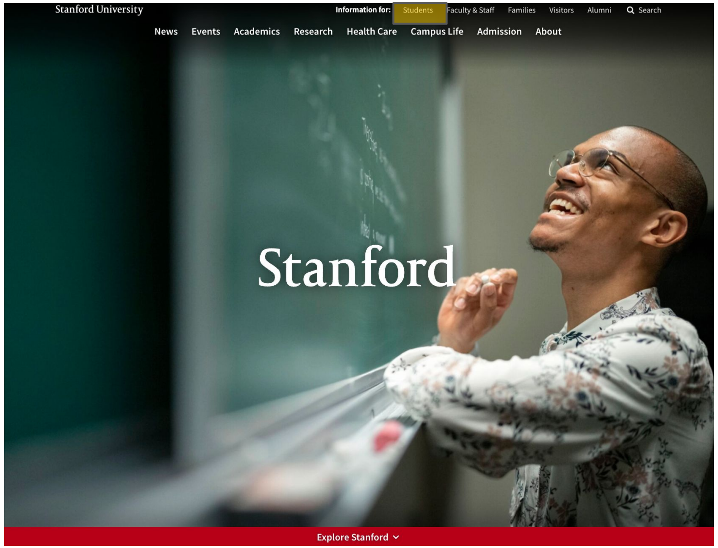


Oh no, it is *“https://www.stanford.edu/”* again -- we had a bug in the code!

Search Engine: Web Crawler - Recursion

```
class Spider:
    name = 'icme_spider'
    start_urls = 'https://icme.stanford.edu/'
    parsed_urls = []

    def parse(self, url: str):
        self.parse_url.append(url)
        for next_url in Website(url):
            if next_url not in self.parsed_urls:
                self.parse(next_url)
```

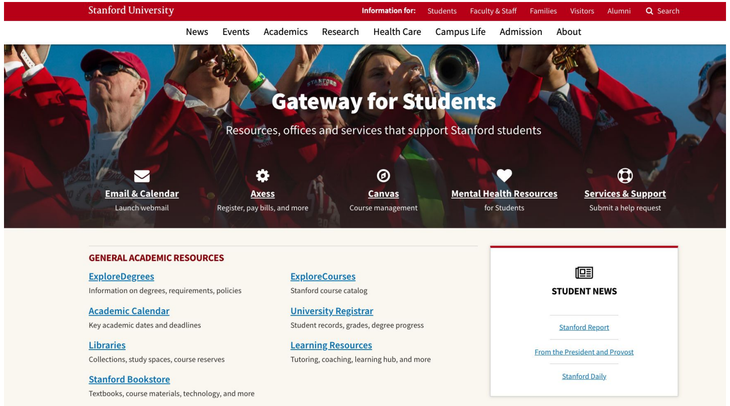
parsed_urls = [“https://icme.stanford.edu”,
“https://www.stanford.edu/”
]



Search Engine: Web Crawler - Recursion

```
class Spider:
    name = 'icme_spider'
    start_urls = 'https://icme.stanford.edu/'
    parsed_urls = []

    def parse(self, url: str):
        self.parse_url.append(url)
        for next_url in Website(url):
            if next_url not in self.parsed_urls:
                self.parse(next_url)
```

parsed_urls = [“https://icme.stanford.edu”,
“https://www.stanford.edu/”
“https://www.stanford.edu/student-gateway/”
]



Search Engine: Full-text Search

- It has a search bar!
  - **-> How to incorporate the user intention?**
- The items are mal-defined at the first glance.
  - -> How to crawl the internet and store the items?
- User does not simply rate the search results!
  - -> How to assign user-item rating with user data?

Search Engine: String Search

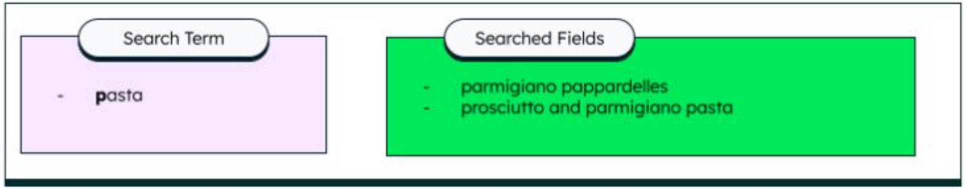


Diagram: https://www.mongodb.com/basics/full-text-search

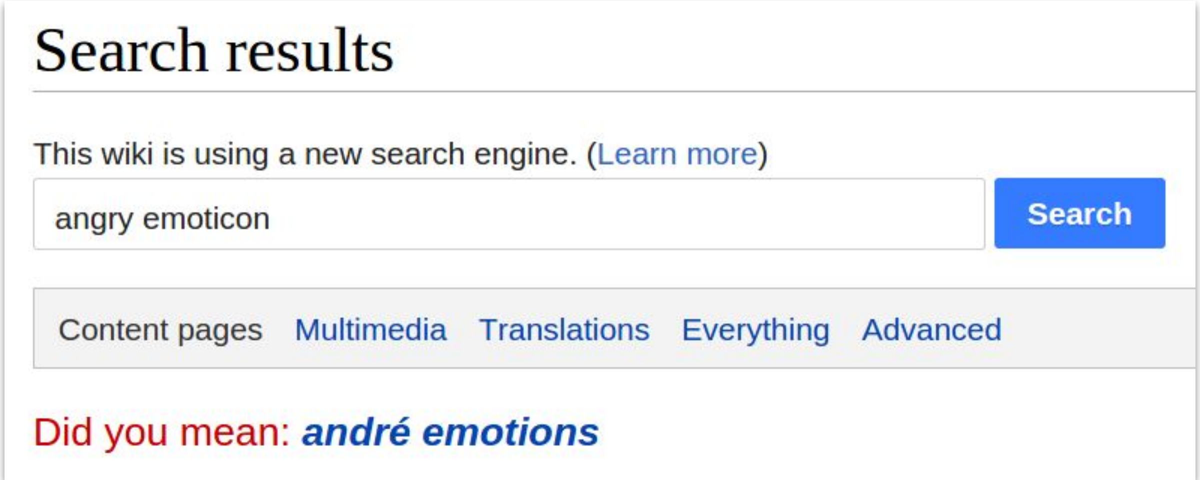
Average: O(n+m); Worst case: O(mn)

Rabin-Karp algorithm, which looks for matching substrings, is fast and easy to implement.
Knuth-Morris-Pratt algorithm looks for all instances of a matching character, increasing the speed for multiple matches in a string.

Mentimeter

Join at **menti.com** use code **1265 3592**

Search Engine: Fuzzy Search



- insertion: *cot* → *coat*
- deletion: *coat* → *cot*
- substitution: *coat* → *cost*

Search Engine: Full-text Search - Inverted Index

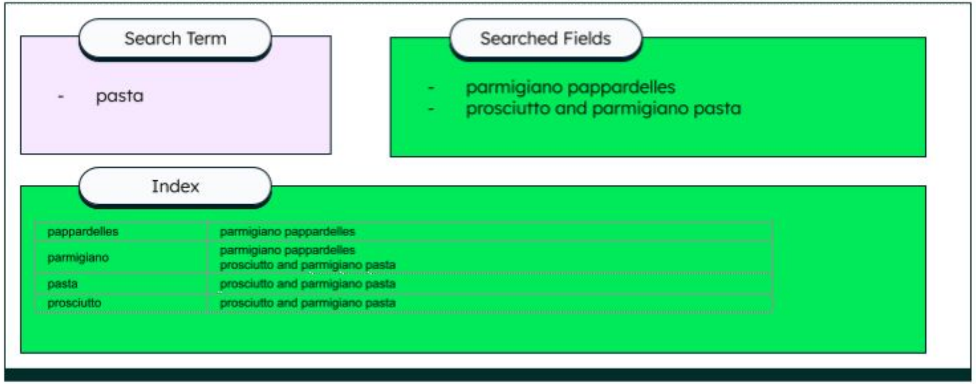


Diagram: https://www.mongodb.com/basics/full-text-search

Search Engine: Full-text Search - Inverted Index

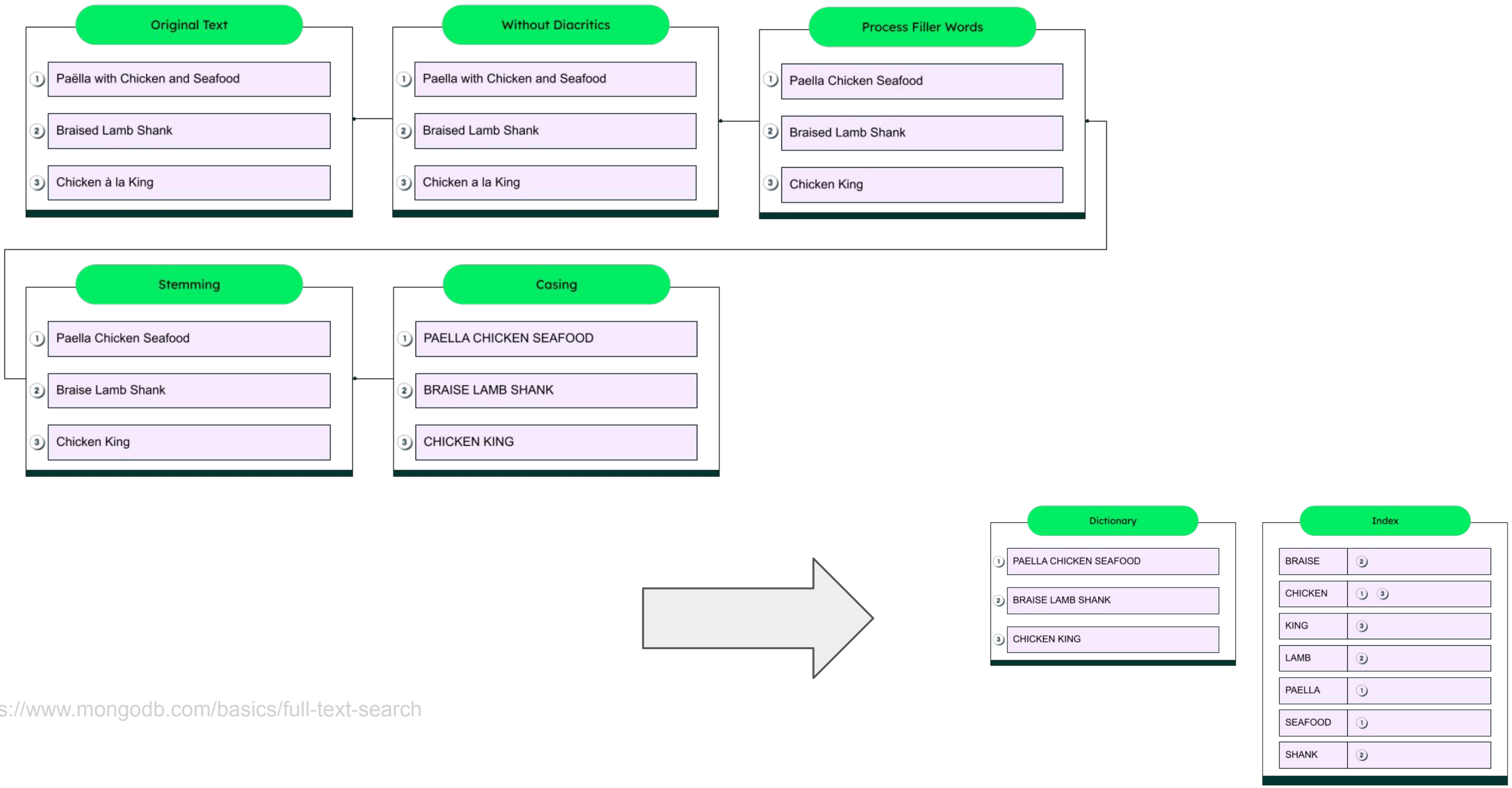


Diagram: https://www.mongodb.com/basics/full-text-search

Search Engine: User-item Rating

- It has a search bar!
  - -> How to incorporate the user intention?
- The items are mal-defined at the first glance.
  - -> How to crawl the internet and store the items?
- User does not simply rate the search results!
  - -> How to assign user-item rating with user data?

Search Engine: PageRank

- So far, web pages are treated as individual documents.
- But there are *hyperlinks* between them!

Search Engine: PageRank

$$PR(p_i) = \frac{1-d}{N} + d \sum_{p_j \in M(p_i)} \frac{PR(p_j)}{L(p_j)}$$

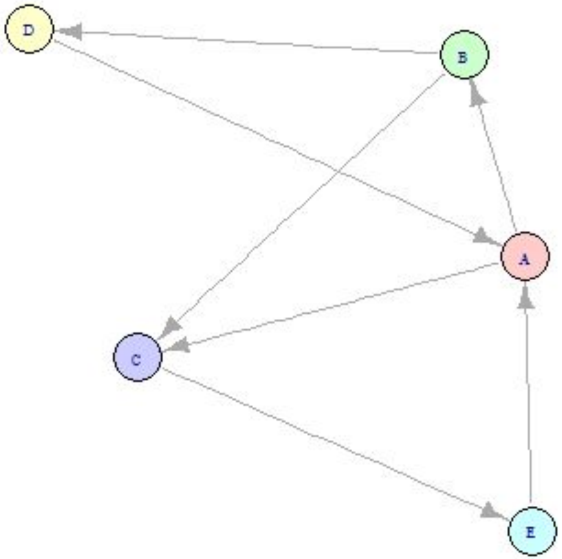
$p_1, p_2, \ldots, p_N$ are the pages under consideration

$M(p_i)$ is the set of pages that link to p_i

$L(p_j)$ is the number of outbound links on page p_j

Page Rank of the nodes at start

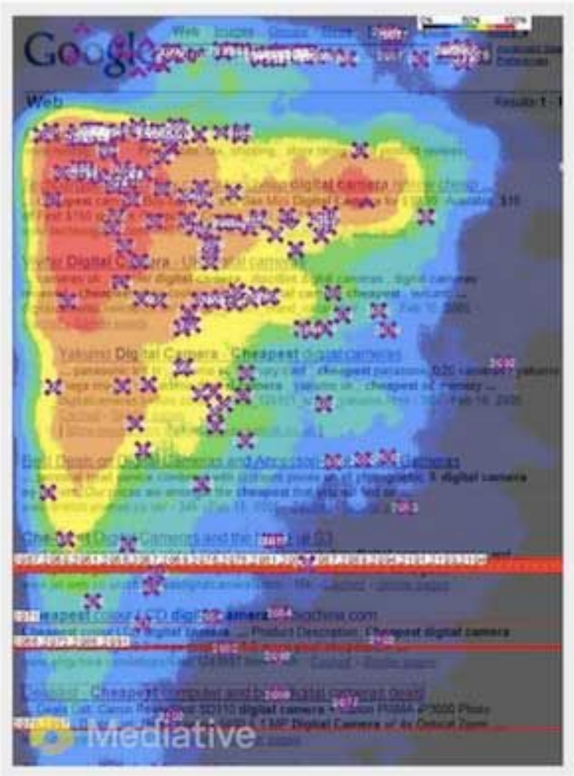
	Rank
A	0.2
B	0.2
C	0.2
D	0.2
E	0.2



Search Engine: User-item Rating from User Behavior

- We have millions of user clicking on some Search Engine Results every day.
- Can we assign click or not as ratings?
  - Yes and no

Search Engine: User-item Rating - Clicks

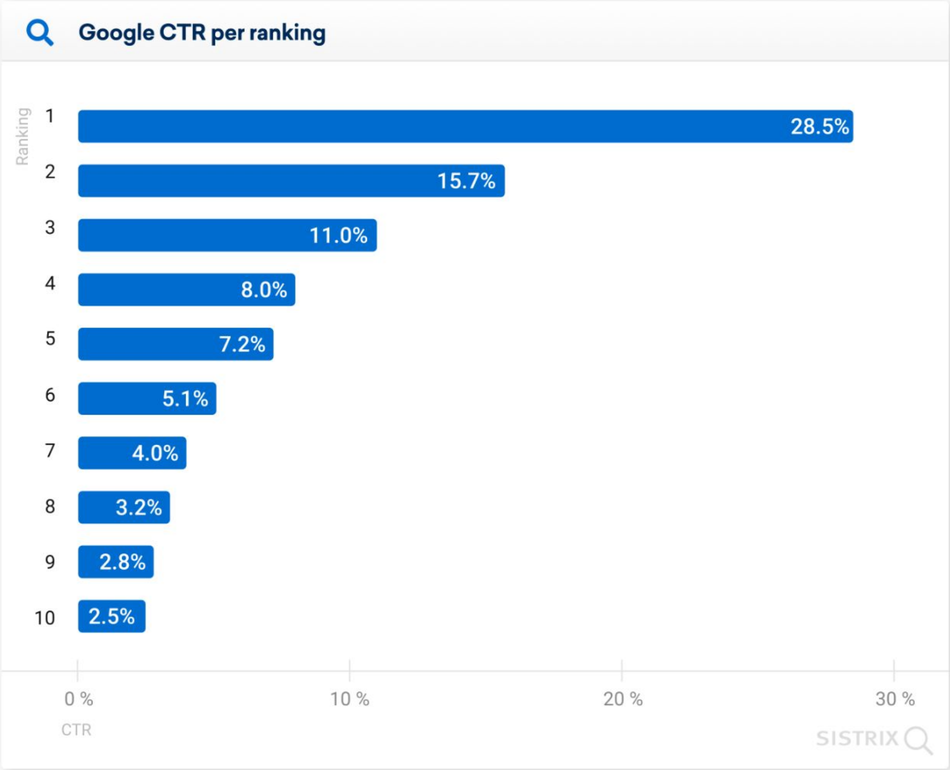


2005

2014

Source: The Evolution of Google Search Results Pages, Mediative, 2014

Search Engine: User-item Rating - Clicks



Search Engine: Evaluation - Clicks

$$\text{CTR} = \frac{\text{CLICKS (Number of people who clicked the ad)}}{\text{IMPRESSIONS (Number of people who saw the ad)}} \times 100$$

Search Engine: Recap of Online Evaluation

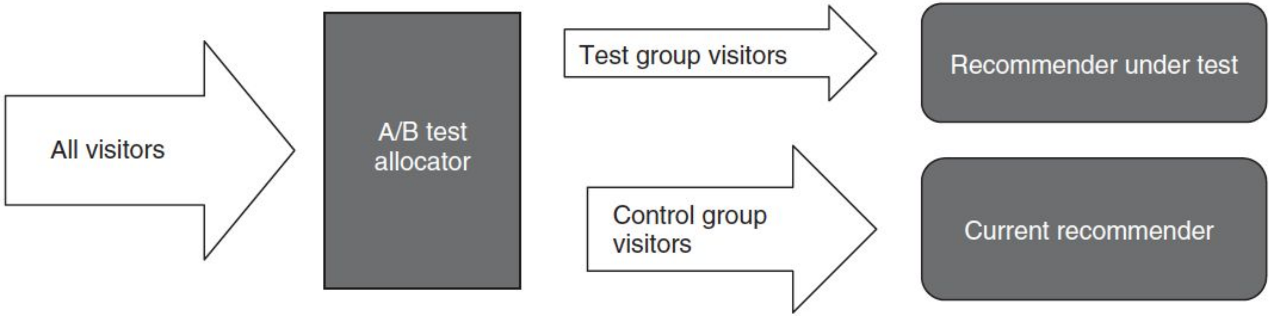


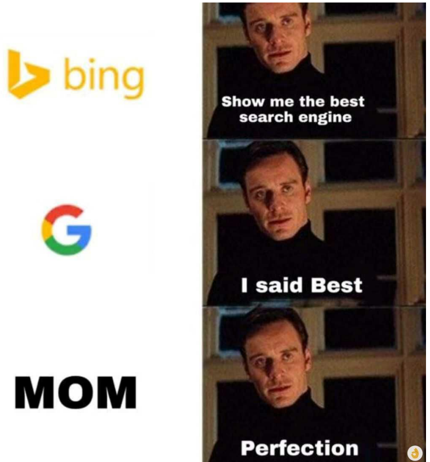
Figure 9.16 In an A/B test, visitors are split into two groups: the test group that sees the new feature and a control group that continues as usual.

Search Engine: How to measure user satisfaction and success?

- **Q:** For Search Engine, how to measure the user satisfaction and success?
  - Any potential bias?
- **Q:** How about the recommendation system you chose yesterday?

Search Engine: Time-to-long-click (TTLC)

- Long-click: When a user performs a search, clicks through on a result and remains on that site for a long time.
    - Anti-pattern: Pogo-sticking
- Domain specific
- Knowledge panels and direct answers



Tiktok: Multi-target

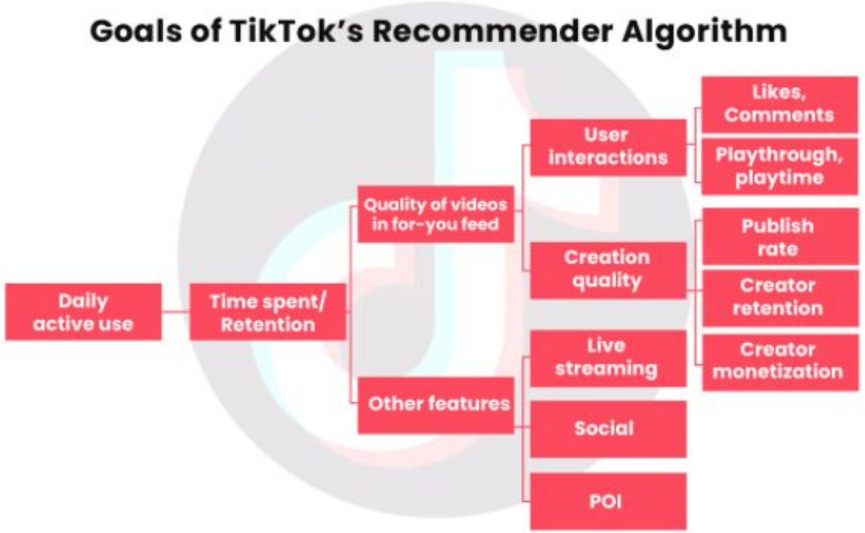
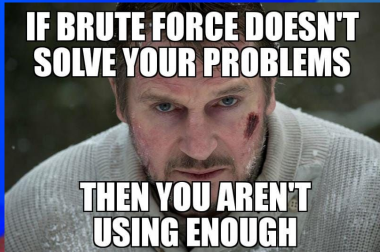


Diagram:
https://www.linkedin.com/pulse/ai-behind-tiktoks-addictive-algorithm-simple-alexander-stahl/

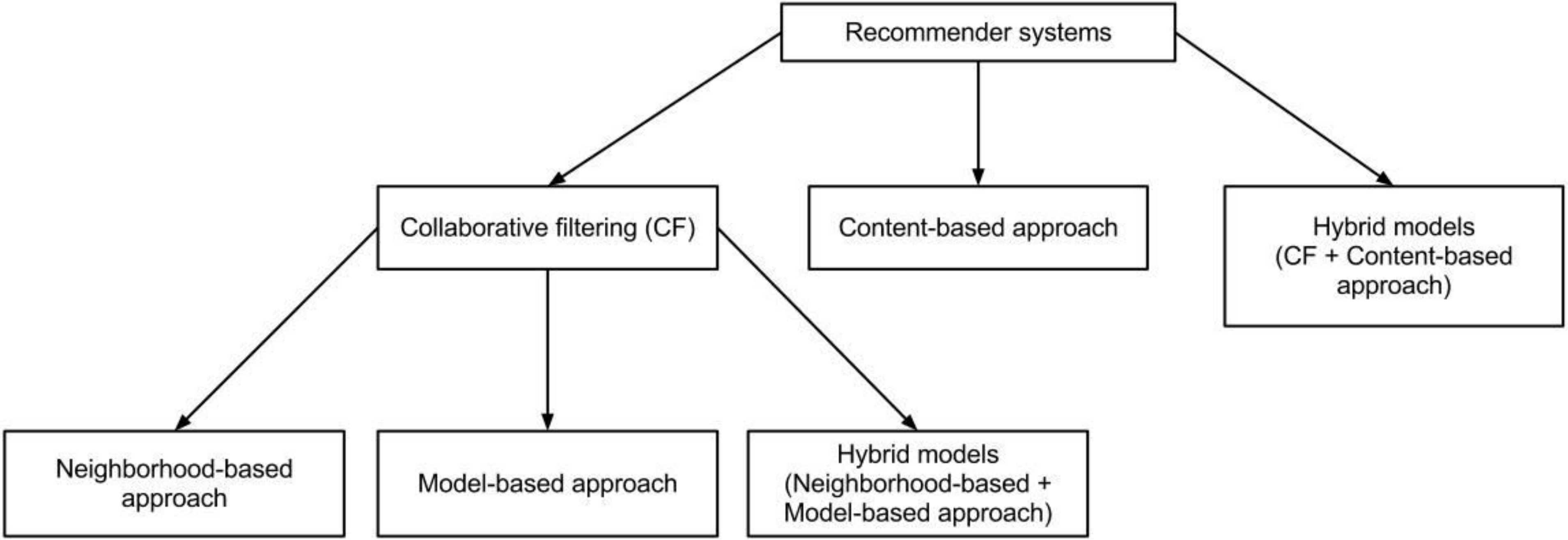
Advanced Topics of Recommendation System



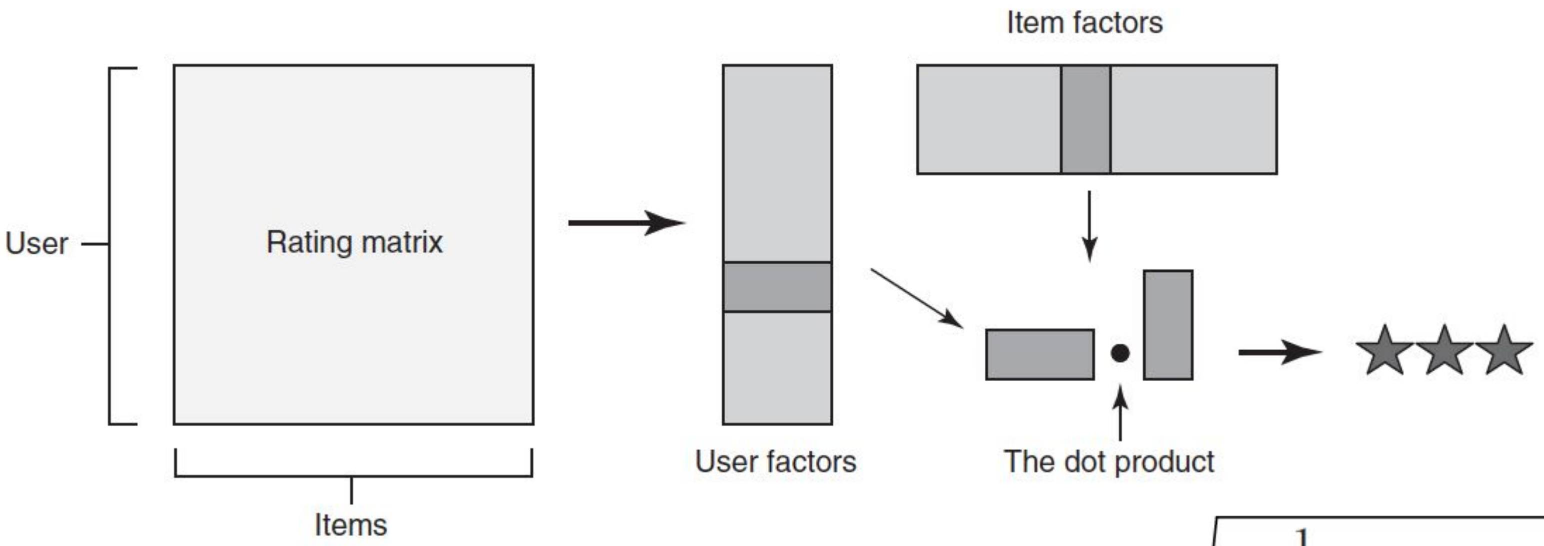
Advanced Topics of Recommendation System

- Deep learning
- Scale up and speed up
- LLM + Recommendation system
- Social impact

Recommendation System w/ Deep Learning: Recap



Recommendation System w/ Deep Learning: Recap

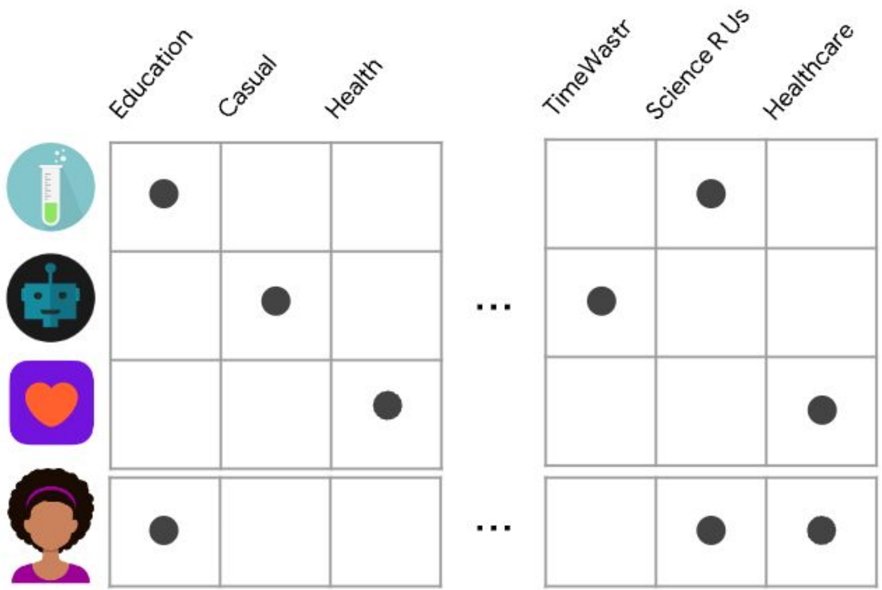


$$RMSE = \sqrt{\frac{1}{|known|} \sum_{(u,i) \in known} (r_{ui} - u_u v_i)^2}$$

- Rating/score can be modeled as a product of user vector and item vector.

Recommendation System w/ Deep Learning: Recap

Example from: https://developers.google.com/machine-learning/recommendation/content-based/basics



- There are well-defined features (of both users and items) that can be used for the prediction.

RS w/ Deep Learning: Factorization Machines (2010)

- Factorization Machines (FM): Let’s combine the best of the both worlds!

Reformulate Collaborative Filtering

	i_1	i_2	i_3
u_1	2	4	
u_2		1	
u_3	3		5

							y
$x^{(0)}$	1	0	0	1	0	0	2
$x^{(1)}$	1	0	0	0	1	0	4
$x^{(2)}$	0	1	0	0	1	0	1
$x^{(3)}$	0	0	1	1	0	0	3
$x^{(4)}$	0	0	1	0	0	1	5

Users

Items

Observed Ratings

Reformulate Collaborative Filtering

	i_1	i_2	i_3
u_1	2	4	
u_2		1	
u_3	3		5

$$\hat{r}_{1,1} = w_0 + u_1^T v_1$$

	Users			Items			y
$x^{(0)}$	1	0	0	1	0	0	2
$x^{(1)}$	1	0	0	0	1	0	4
$x^{(2)}$	0	1	0	0	1	0	1
$x^{(3)}$	0	0	1	1	0	0	3
$x^{(4)}$	0	0	1	0	0	1	5

Observed Ratings

$$\hat{r}_{1,1} = w_0 + \left(\begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} (u_1, u_2, u_3) \right)^T \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} (v_1, v_2, v_3)$$

Reformulate Collaborative Filtering

	i_1	i_2	i_3
u_1	2	4	
u_2		1	
u_3	3		5

$$\hat{r}_{1,1} = w_0 + u_1^T v_1$$

	Users			Items			y (Observed Ratings)
$x^{(0)}$	1	0	0	1	0	0	2
$x^{(1)}$	1	0	0	0	1	0	4
$x^{(2)}$	0	1	0	0	1	0	1
$x^{(3)}$	0	0	1	1	0	0	3
$x^{(4)}$	0	0	1	0	0	1	5

$$\hat{r}_{1,1} = w_0 + \sum_{i=1}^{n} \sum_{j=i+1}^{n} x_i^{(0)} x_j^{(0)} u_i^T v_j$$

Reformulate Content Filtering

	i_1	i_2	i_3
u_1	2	4	
u_2		1	
u_3	3		5

E.g. News Popularity

	i_1	i_2	i_3	a_1	a_2	y
$x^{(0)}$	1	0	0	2.0	0.0	2
$x^{(1)}$	0	1	0	1.5	0.5	4
$x^{(2)}$	0	1	0	1.5	0.5	1
$x^{(3)}$	1	0	0	2.0	0.0	3
$x^{(4)}$	0	0	1	3.2	1.7	5

Observed Ratings

Items

Auxiliary Features

$$\hat{r}_{1,1} = w_0 + w_1 a_1 + w_2 a_2$$

Back to Factorization Machines (FM)

	i_1	i_2	i_3
u_1	2	4	
u_2		1	
u_3	3		5

	Users			Items			Auxiliary Features		Observed Ratings
$x^{(0)}$	1	0	0	1	0	0	2.0	0.0	2
$x^{(1)}$	1	0	0	0	1	0	1.5	0.5	4
$x^{(2)}$	0	1	0	0	1	0	0.0	1.0	1
$x^{(3)}$	0	0	1	1	0	0	0.3	0.7	3
$x^{(4)}$	0	0	1	0	0	1	3.2	1.7	5

$$\hat{r} = w_0 + \sum_{i=1}^{n} w_i x_i + \sum_{i=1}^{n} \sum_{j=i+1}^{n} x_i x_j v_i^T v_j$$

Back to Factorization Machines (FM)

$$\hat{r} = w_0 + \sum_{i=1}^{n} w_i x_i + \sum_{i=1}^{n} \sum_{j=i+1}^{n} x_i x_j v_i^T v_j$$

- w_0: Global bias
- w_i: Weights of features
- v_i: Feature factor i

=> **Supervised learning paradigm!!**

Back to Factorization Machines (FM)

$$\hat{r} = w_0 + \sum_{i=1}^{n} w_i x_i + \sum_{i=1}^{n} \sum_{j=i+1}^{n} x_i x_j v_i^T v_j$$

- w_0: Global bias
- w_i: Weights of features
- v_i: Feature factor i

=> Supervised learning paradigm!!

Rating Matrix

Collaborative Filtering

FM

Welcome to the Deep-world

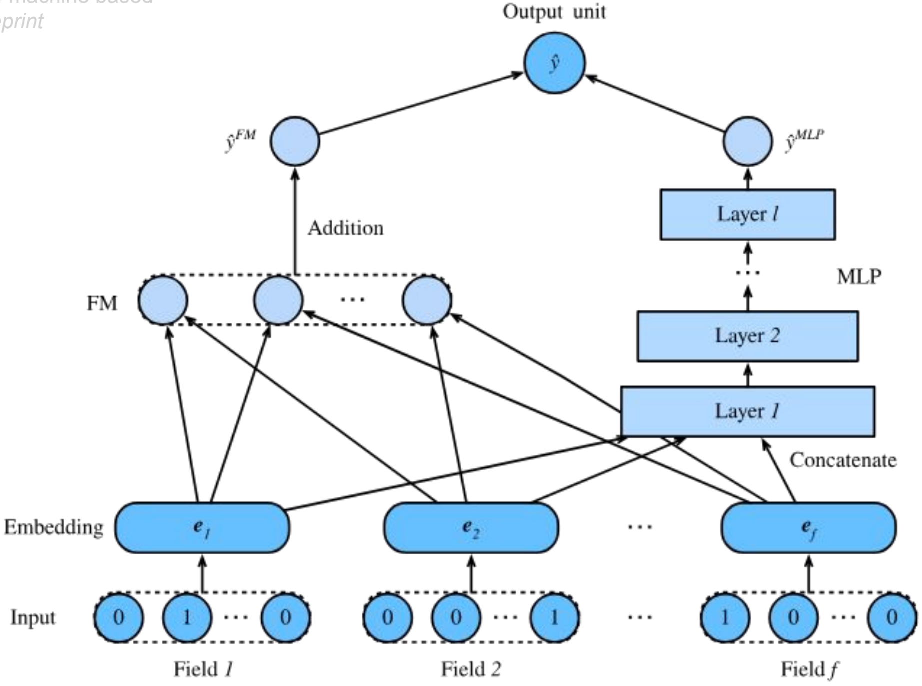
Factorization Machines (FM): Pros & Cons

- Capture user-item feature interaction
- Efficient for sparse data

- Non-linear patterns

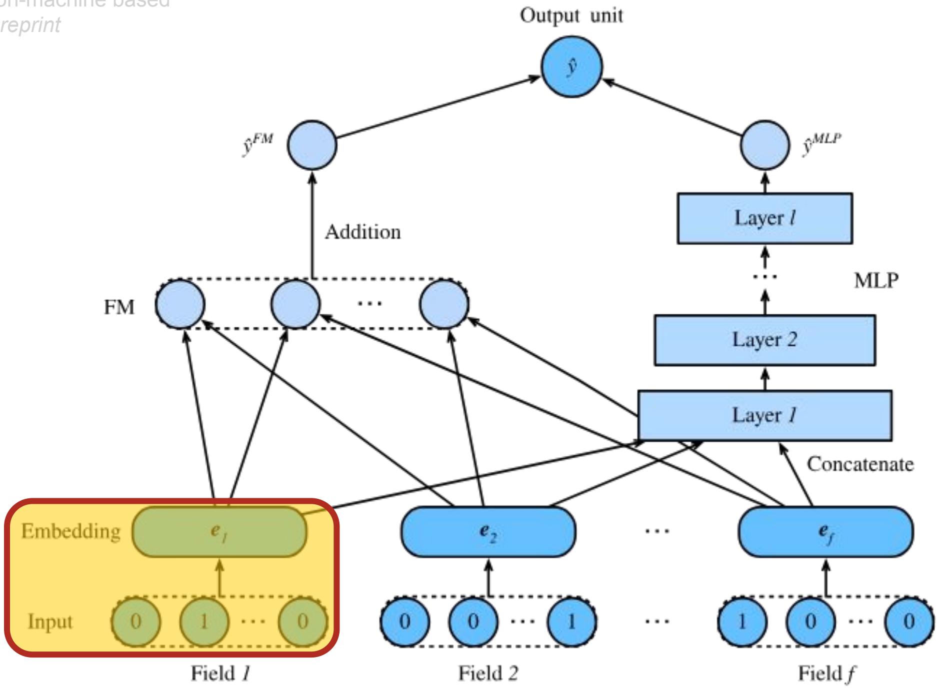
DeepFM (2017)

Guo, Huifeng, et al. "DeepFM: a factorization-machine based neural network for CTR prediction." *arXiv preprint arXiv:1703.04247* (2017).



DeepFM (2017)

Guo, Huifeng, et al. "DeepFM: a factorization-machine based neural network for CTR prediction." *arXiv preprint arXiv:1703.04247* (2017).

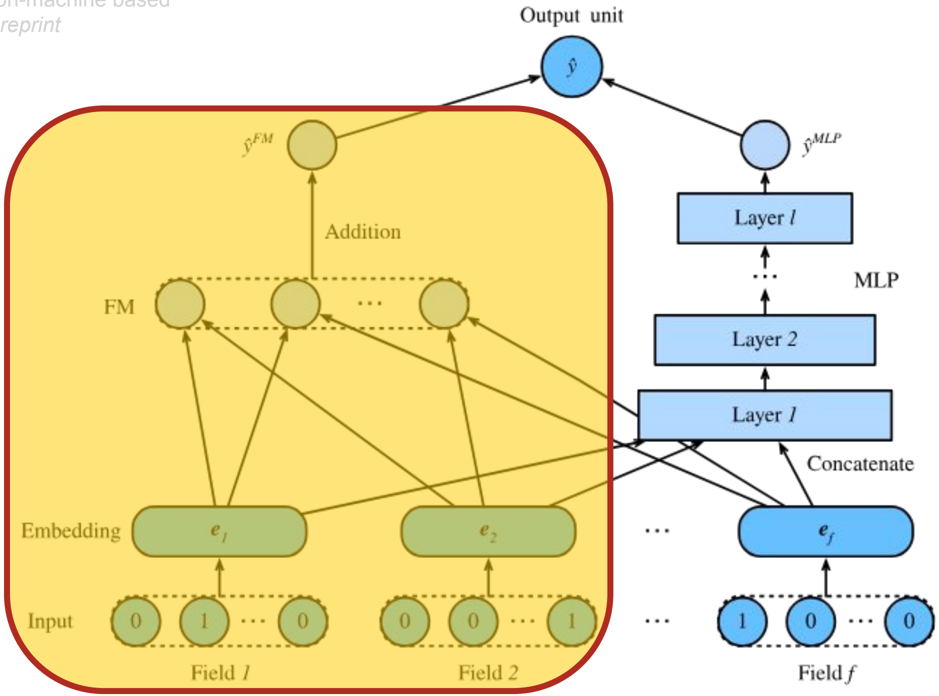


Factors = Embeddings

DeepFM (2017)

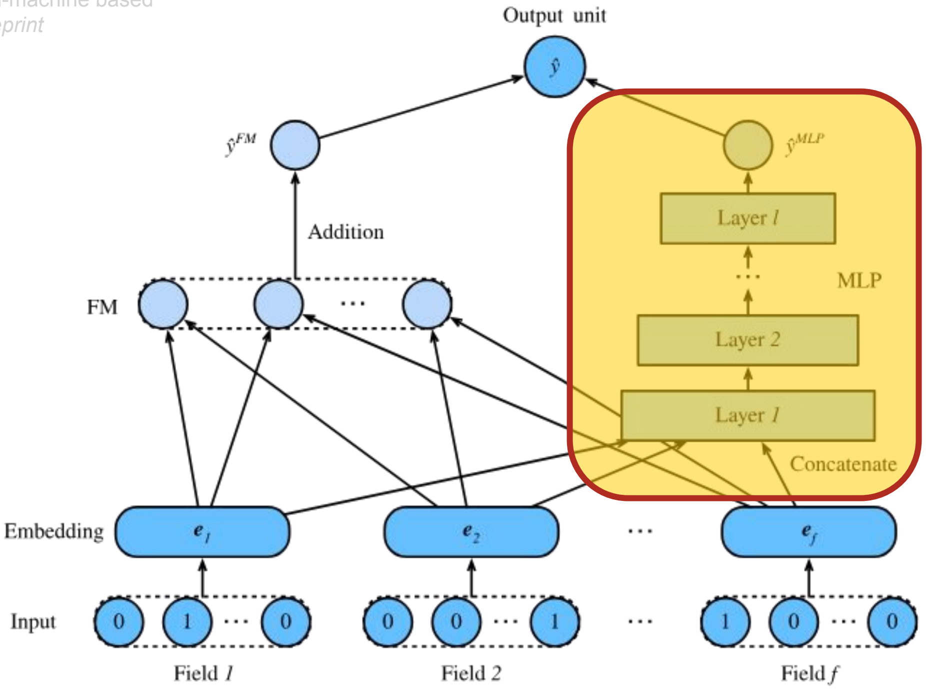
Guo, Huifeng, et al. "DeepFM: a factorization-machine based neural network for CTR prediction." *arXiv preprint arXiv:1703.04247* (2017).

Factorization Machines



DeepFM (2017)

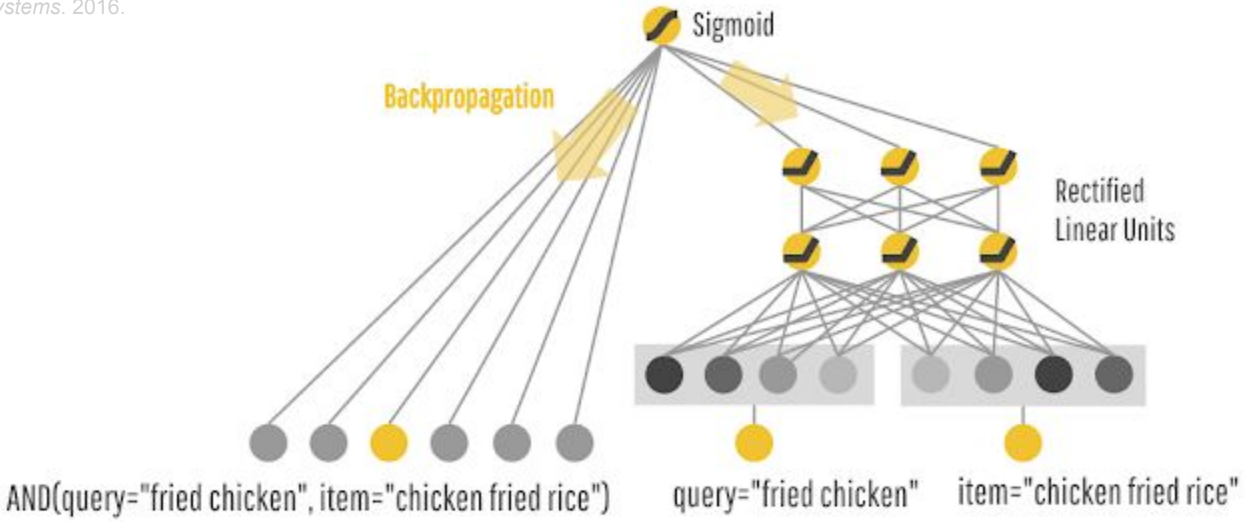
Guo, Huifeng, et al. "DeepFM: a factorization-machine based neural network for CTR prediction." *arXiv preprint arXiv:1703.04247* (2017).



Deep MLP

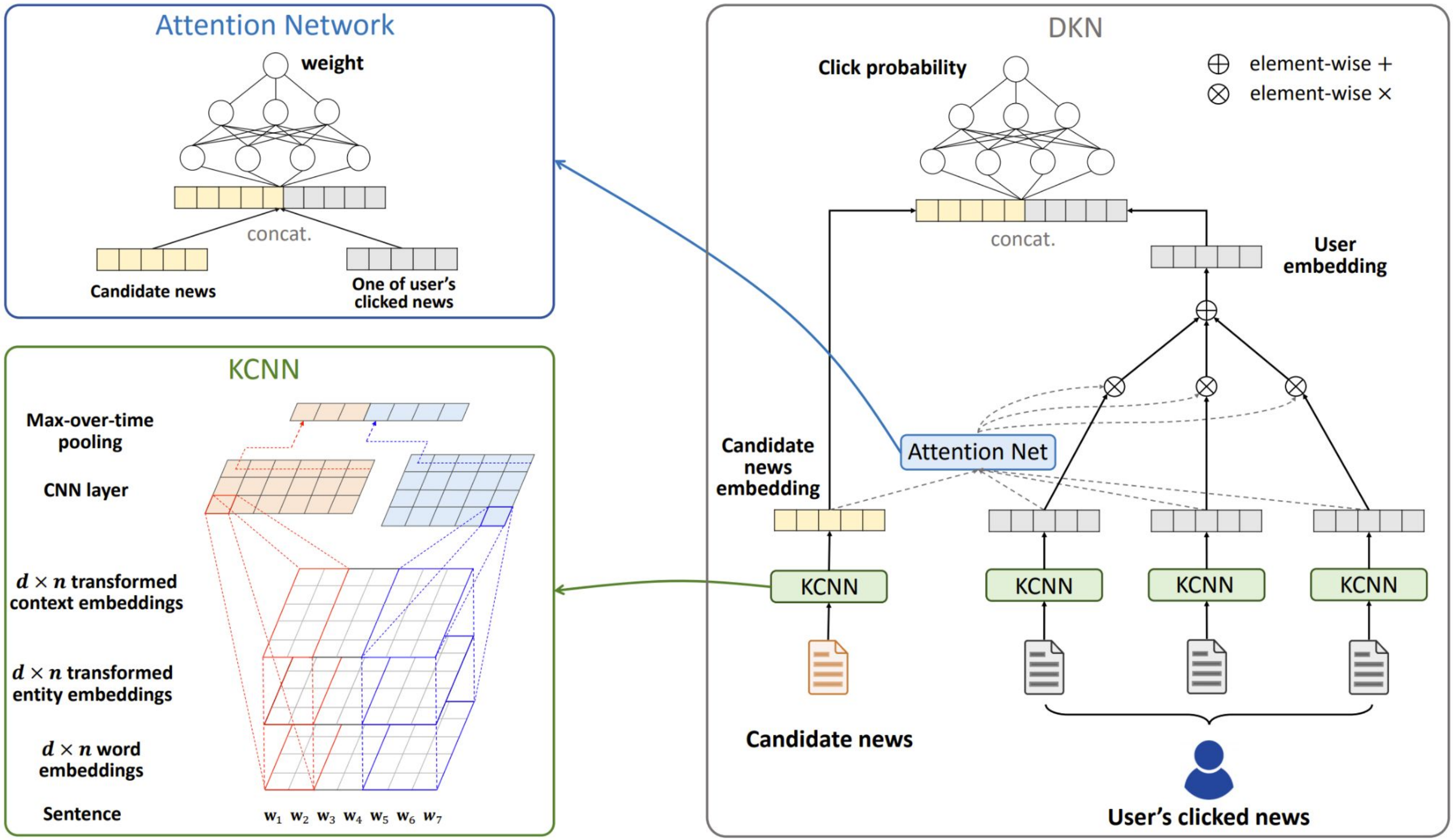
Wide & Deep (2016)

Cheng, Heng-Tze, et al. "Wide & deep learning for recommender systems." *Proceedings of the 1st workshop on deep learning for recommender systems.* 2016.



Deep Knowledge-Aware Network (2018)

Wang, Hongwei, et al. "DKN: Deep knowledge-aware network for news recommendation." *Proceedings of the 2018 world wide web conference*. 2018.



Feature Engineering for Deep Recommendation

- Sparse Features: Tag, device, etc.
- Dense Features: Age, income, # of videos watched, etc.
- **Sparse Features** are the first-class citizens.

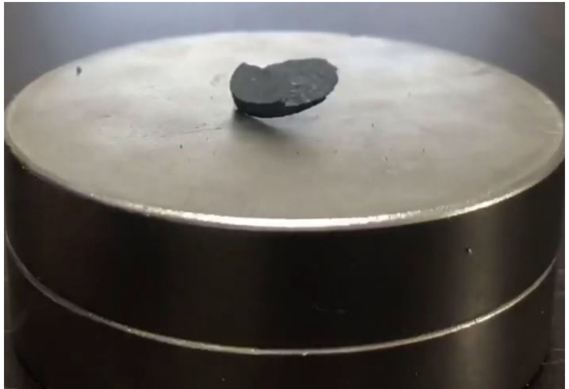
Feature Engineering for Deep Recommendation

- **Sparse Features** are the first-class citizens.
  - Dense feature requires more parameterization.
    - Age -> like this YouTube Video
      - Assumption: Age = 9 -> Age = 10 has a same effect of Age = 43 -> Age = 44

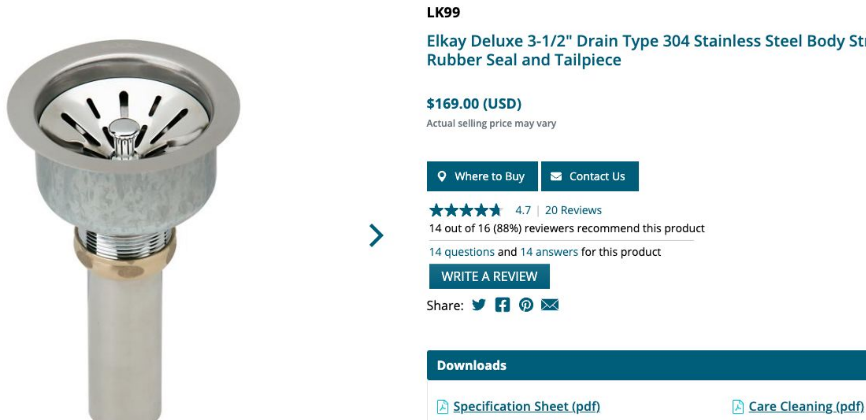
Feature Engineering for Deep Recommendation

- **Sparse Features** are the first-class citizens.
    - Dense feature requires more parameterization.
    - Sparse features are easy to generate cross-features.
        - Hashing tricks: Only keep K more frequent tuples of the combinations.

#lk99 #superconductor



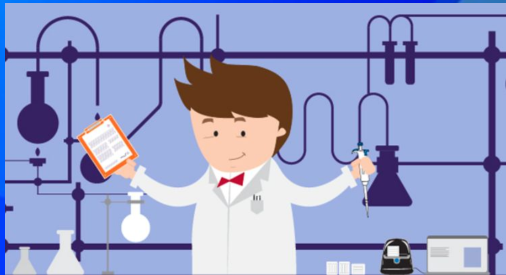
#lk99 #drinking-solution



Feature Engineering for Deep Recommendation

- **Sparse Features** are the first-class citizens.
  - Dense feature requires more parameterization.
  - Sparse features are easy to generate cross-features.
  - * Easy for online training/serving

Lab Time!



Advanced Topics (Part II)

- Deep learning
- **Scale up and speed up**
- LLM + recommendation system
- Social impact

Scale up & Speed up: Speed Matters

*All other things being equal ... our experiments demonstrate that slowing down the search results page by **100 to 400** milliseconds has a measurable impact on the number of searches per user of **-0.2%** to **-0.6%**. That's 0.2% to 0.6% fewer searches for changes under half a second!*

-- Google Research Blog

Speed-up: Recommendation Funnel

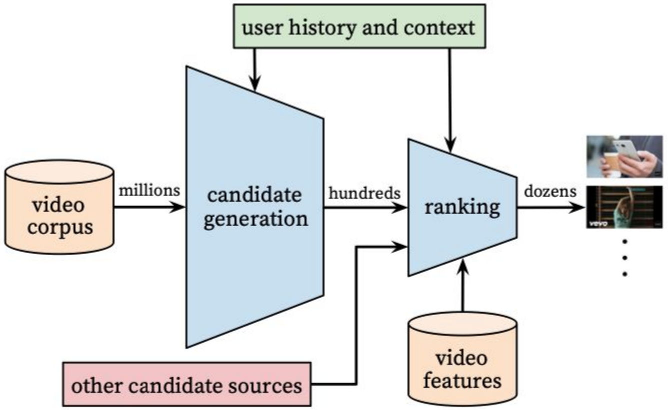
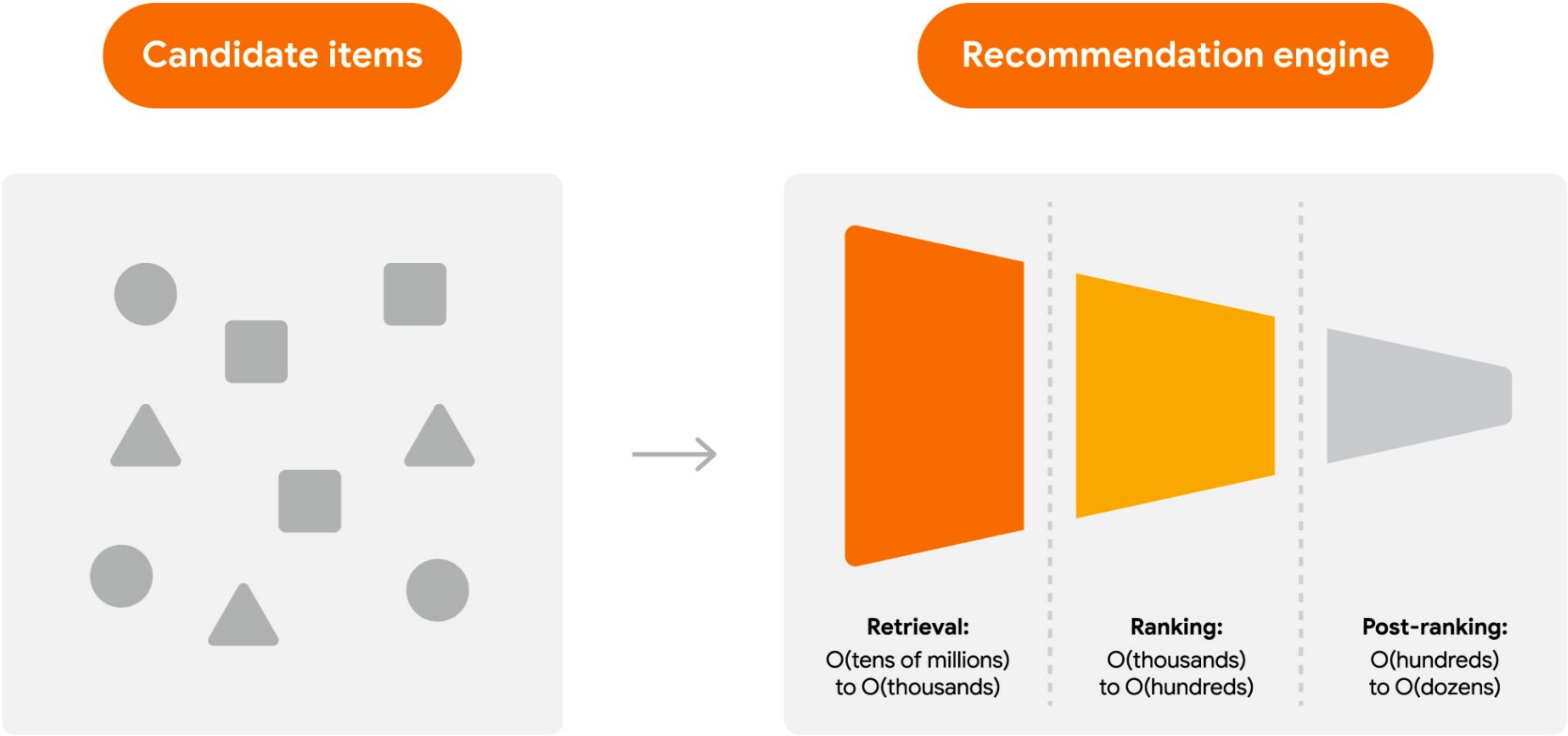


Figure 2: Recommendation system architecture demonstrating the "funnel" where candidate videos are retrieved and ranked before presenting only a few to the user.

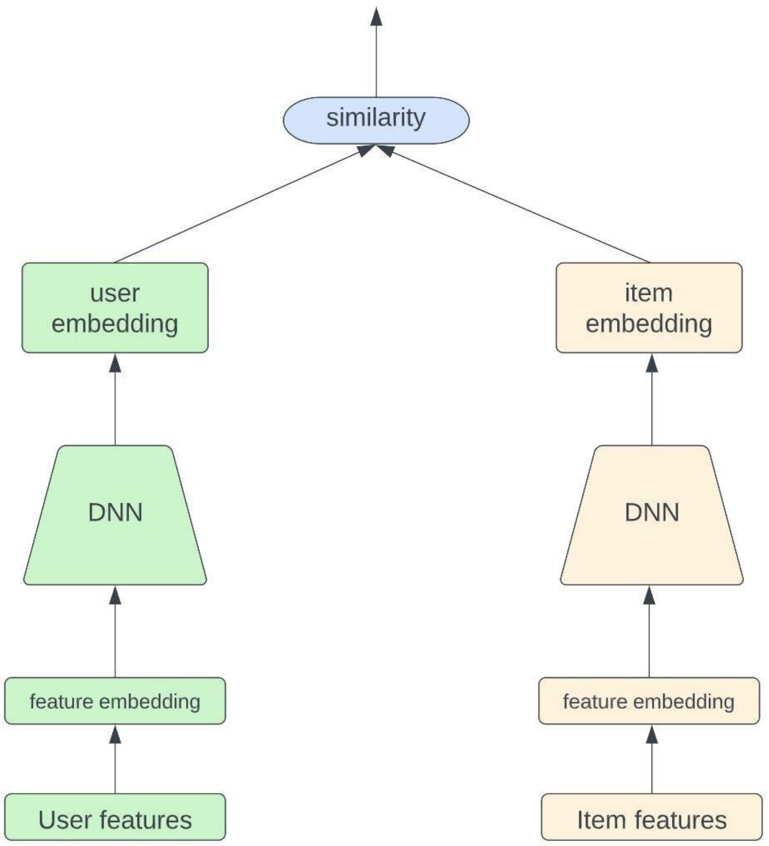
- Retrieval / Candidate-gen
- Ranking / Sort
- Re-rank

Covington, Paul, Jay Adams, and Emre Sargin. "Deep neural networks for youtube recommendations." Proceedings of the 10th ACM conference on recommender systems. 2016.

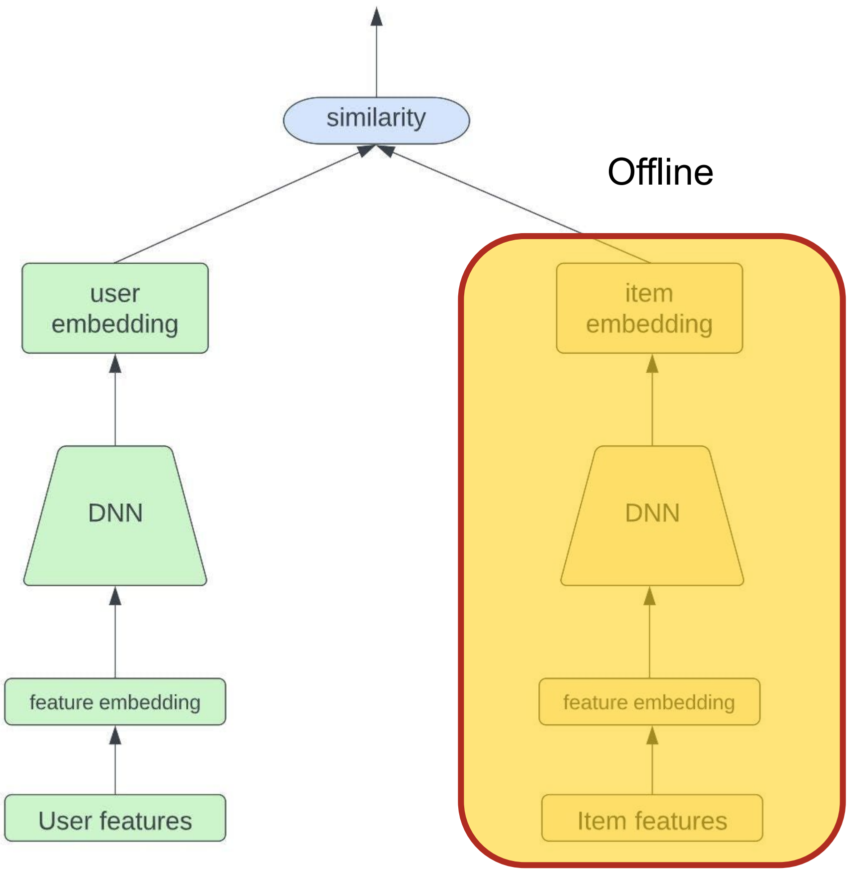
Speed-up: Recommendation Funnel

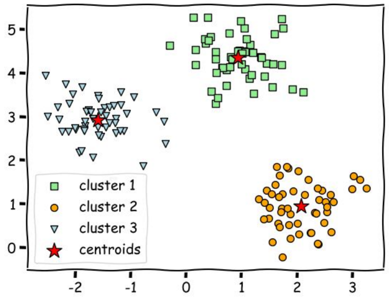


Speed-up: Two-tower Retrieval

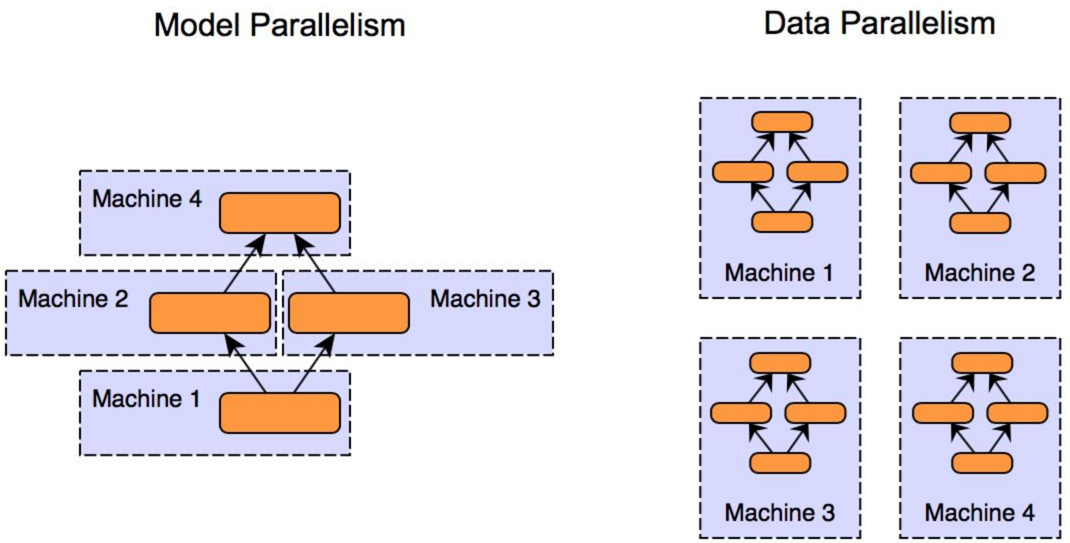


Speed-up: Two-tower Retrieval

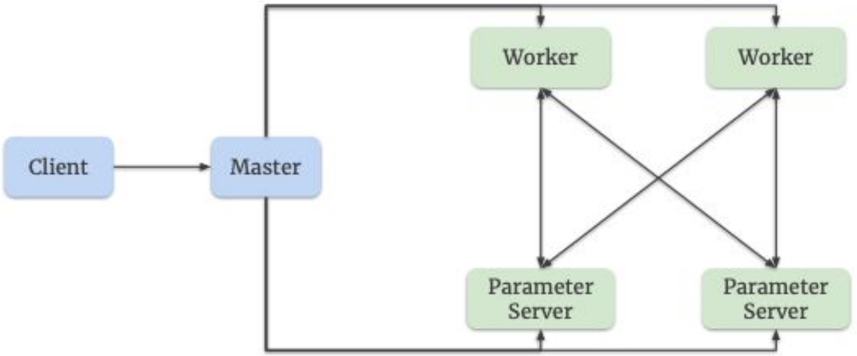




Scale-up: Parameter Server (PS)



Scale-up: Parameter Server (PS)



K, V
storage

Algorithm 1 Distributed Subgradient Descent

Task Scheduler:

issue LoadData() to all workers
for iteration $t = 0, \ldots, T$ **do**
    issue $\textsc{WorkerIterate}(t)$ to all workers.
end for

Worker $r = 1, \ldots, m$:

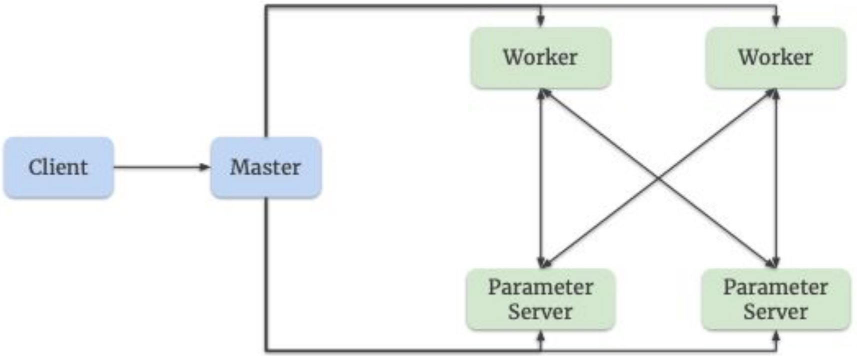
function $\textsc{LoadData}()$
    load a part of training data $\{y_{i_k}, x_{i_k}\}_{k=1}^{n_r}$
    pull the working set $w_r^{(0)}$ from servers
end function
function $\textsc{WorkerIterate}(t)$
    gradient $g_r^{(t)} \leftarrow \sum_{k=1}^{n_r} \partial\ell(x_{i_k}, y_{i_k}, w_r^{(t)})$
    push $g_r^{(t)}$ to servers
    pull $w_r^{(t+1)}$ from servers
end function

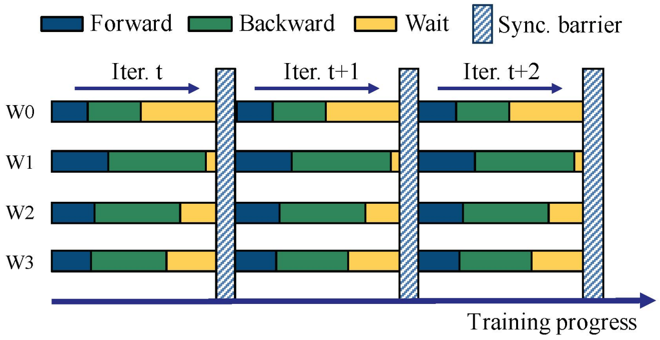
Servers:

function $\textsc{ServerIterate}(t)$
    aggregate $g^{(t)} \leftarrow \sum_{r=1}^{m} g_r^{(t)}$
    $w^{(t+1)} \leftarrow w^{(t)} - \eta\left(g^{(t)} + \partial\Omega(w^{(t)})\right)$
end function

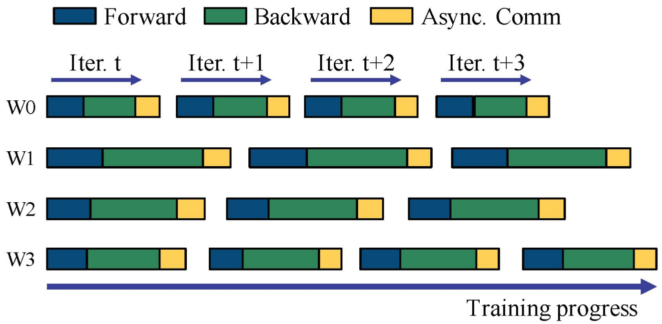
Li, Mu, et al. "Communication efficient distributed machine learning with the parameter server." Advances in Neural Information Processing Systems 27 (2014).

Scale-up: Parameter Server (PS)





(a)



(b)

Li, Mu, et al. "Communication efficient distributed machine learning with the parameter server." Advances in Neural Information Processing Systems 27 (2014).

Augmenting recommendation systems with LLMs (Large Language Model)

Large Language Models

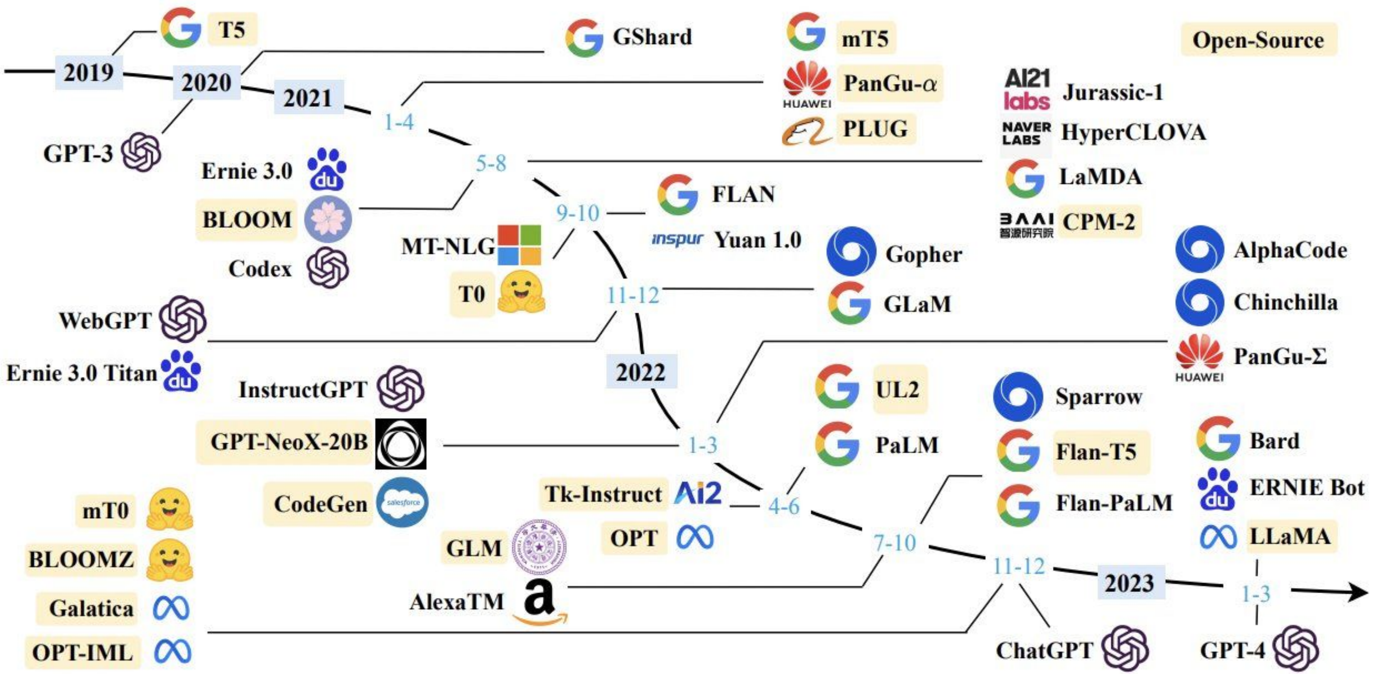
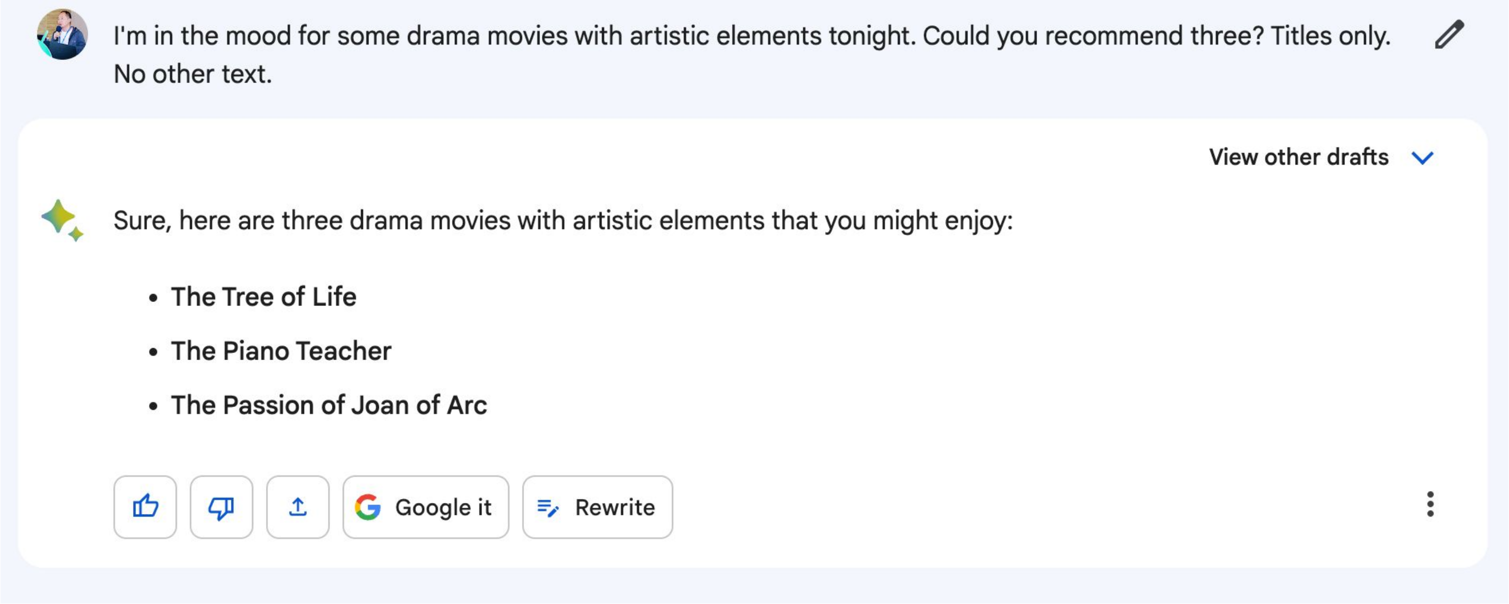


Fig. 1. A timeline of existing large language models (having a size larger than 10B) in recent years. We mark the open-source LLMs in yellow color.

Diagram:
https://www.labellerr.com/blog/overview-of-development-of-large-larnguage-models/

Conversional Recommendation



- Fluid experience
- Limited candidates / High inference cost

Conversional Recommendation: Prompt Engineering

```
prompt = """You are a movie recommender and your job is to predict
            a user's rating (ranging from 1 to 5, with 5 being the highest)
            on a movie, based on that user's previous ratings.

User 42 has rated the following movies:
"Moneyball" 4.5
"The Martian" 4
"Pitch Black" 3.5
"12 Angry Men" 5

Predict the user's rating on "The Matrix".
Output the rating score only.
Do not include other text.
"""
response = palm.generate_text(model="models/text-bison-001", prompt=prompt)
print(response.result)

# 4.5
```

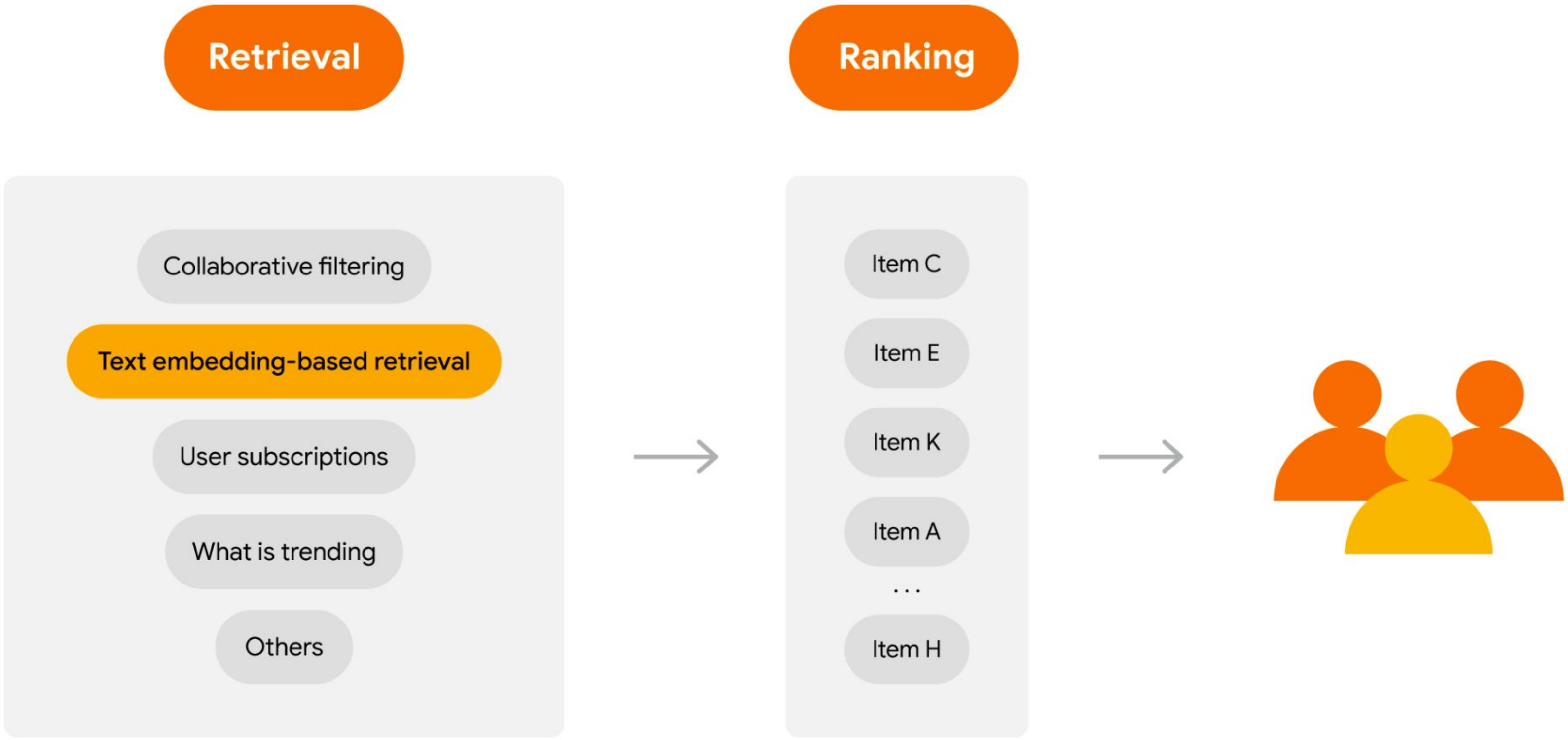
Example from: https://blog.tensorflow.org/2023/06/augmenting-recommendation-systems-with.html

Conversional Recommendation: Prompt Engineering (ranking)

```
prompt = """You are a movie recommender and your job is to recommend new movies
            based on the sequence of movies that a user has watched. You pay special
            attention to the order of movies because it matters.

User 42 has watched the following movies sequentially:

"Margin Call",
“The Big Short”,
"Moneyball",
"The Martian",

Recommend three movies and rank them in terms of priority.
Titles only.
Do not include any other text.
"""

response = palm.generate_text(
   model="models/text-bison-001", prompt=prompt, temperature=0
)
print(response.result)

# 1. The Wolf of Wall Street
# 2. The Social Network
# 3. Inside Job
```

Example from: https://blog.tensorflow.org/2023/06/augmenting-recommendation-systems-with.html

Text embedding-based recommendations



Social Impact

Ethical and Social Issues

- Fairness
  - Unfair/inaccurate recommendation
- Echo chamber
- Privacy

Ethical and Social Issues

- Manipulation
  - Trading and nudging [1]
- Promote Addiction [2]
- Privacy

[1] Burr, C., Cristianini, N., & Ladyman, J. (2018). An Analysis of the Interaction Between Intelligent Software Agents and Human Users. *Minds & Machines, 28*(4), 735–774. Retrieved from https://link.springer.com/article/10.1007/s11023-018-9479-0

[2] Seaver, N. (2018). Captivating algorithms: Recommender systems as traps. *Journal of Material Culture*. Retrieved from https://static1.squarespace.com/static/55eb004ee4b0518639d59d9b/t/5b707506352f5356c8d6e7d2/1534096646595/seaver-captivating-algorithms.pdf

Ethical and Social Issues

- Manipulation
  - Trading and nudging [1]
- Promote Addiction [2]
- Privacy

[1] Burr, C., Cristianini, N., & Ladyman, J. (2018). An Analysis of the Interaction Between Intelligent Software Agents and Human Users. *Minds & Machines, 28*(4), 735–774. Retrieved from https://link.springer.com/article/10.1007/s11023-018-9479-0

[2] Seaver, N. (2018). Captivating algorithms: Recommender systems as traps. *Journal of Material Culture*. Retrieved from https://static1.squarespace.com/static/55eb004ee4b0518639d59d9b/t/5b707506352f5356c8d6e7d2/1534096646595/seaver-captivating-algorithms.pdf

Mentimeter

Join at menti.com use code 1265 3592