

Graduate Certificate in Big Data Analytics

Content-Based Recommender Systems

Dr. Fan Zhenzhen
Institute of Systems Science
National University of Singapore
E-mail: zhenzhen@nus.edu.sg

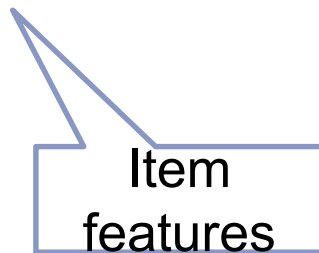
© 2019-2023 NUS. The contents contained in this document may not be reproduced in any form or by any means, without the written permission of ISS, NUS, other than for the purpose for which it has been supplied.

Agenda

- Basics of Content-based Recommender Systems
- Item Representation and Similarity
- Some Example Applications
- Advantages and Issues

Content-based Recommendation

The system learns to recommend items that are **similar** to the ones that the user liked in the past.

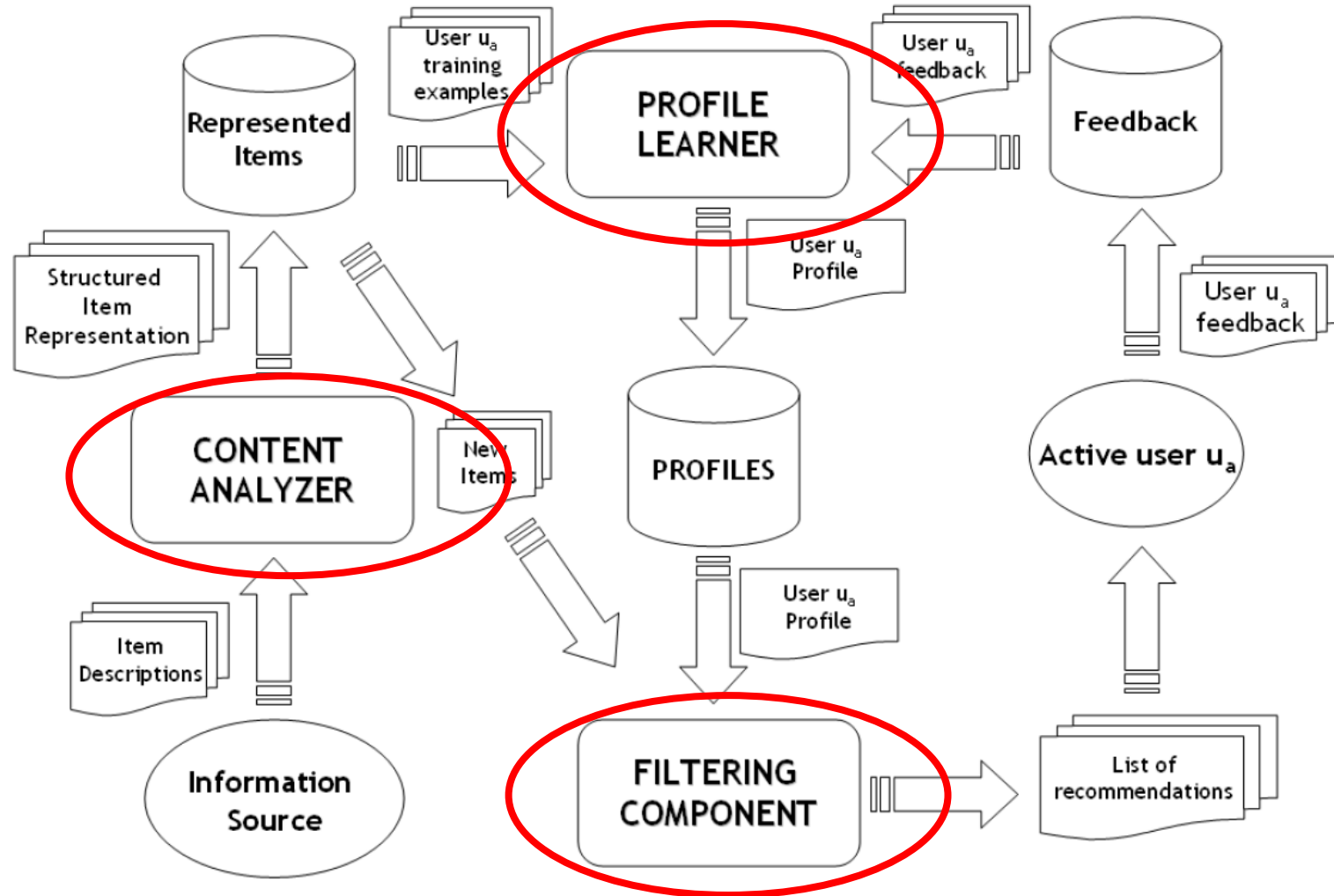


- In contrast, *collaborative* recommendation identifies **users** whose preferences are similar to those of the given user and recommends items they have liked.
- Recommending news, books, movies, music, products and services in e-commerce, etc.

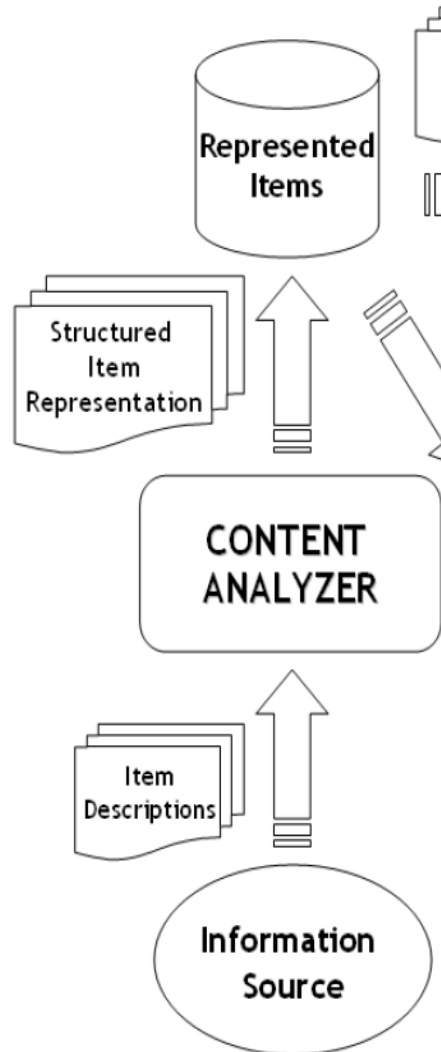
Basics of Content-based Recommendation

- Inputs:
 - Profile of the **user**'s preferences and interests
 - Description of a **content** item
 - Output: a **relevance score** representing the user's level of interest in that object
- ➔ Ranking, filtering of content items

High Level Architecture

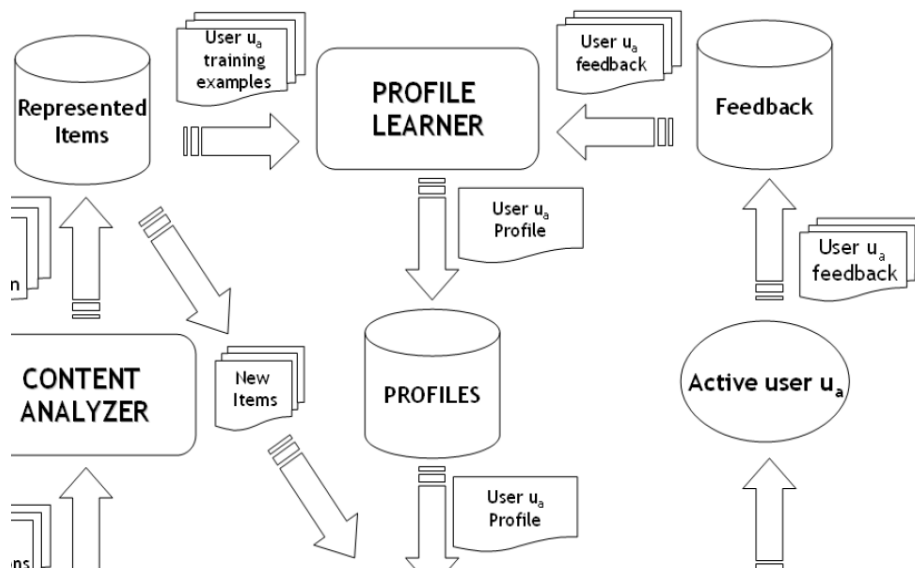


Content Analyzer



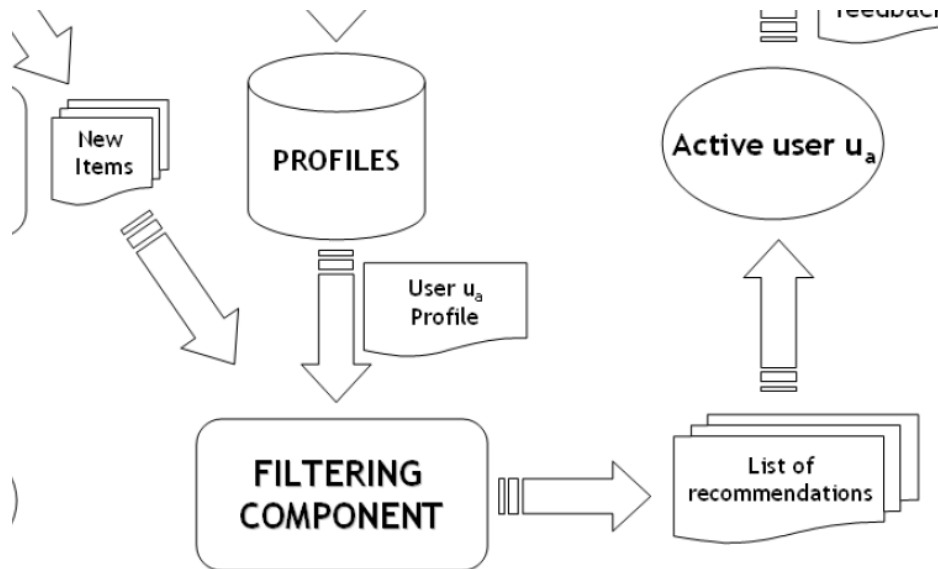
- Input: The content of items (web pages, news, product descriptions, etc)
- Pre-processing step to extract structured relevant information from content -- **feature extraction** to the target space

Profile Learner



- Defined by user
- Or infer a model of user interests from items liked or disliked by the user (machine learning) e.g. relevance feedback method - feedback (training examples) provided by the user

Filtering Component



- Matching user profile against item representations using similarity metrics (eg. Cosine similarity)
- Result: binary or continuous relevance judgement, a (ranked) list of potentially interesting items

User Profile - Feedbacks

- Feedbacks – user’s reactions to items, together with the related item descriptions
 - **Positive** feedbacks – inferring features the user liked
 - **Negative** feedbacks – inferring features the user ‘s not interested in
- **Explicit Feedbacks** – The user explicitly evaluates items.
 - Like/dislike
 - Ratings
 - Text comments
- **Implicit Feedbacks** – no active user involvement; derived by monitoring and analyzing user’s activities
 - Assigning a relevance score to specific user actions on an item (e.g. saving, discarding, printing, bookmarking, etc.)

Item Representation

- A set of *features*, or *attributes*, *properties*.
- Features may be manually assigned, e.g. features for movies - *actors, directors, genres, subject matter*, etc.
- For items like web pages, emails, news articles, and product descriptions, features are extracted from **text – vector space model**
 - Keyword-based profiles
 - Not able to capture the semantics of user interests
 - Language ambiguity like *polysemy* and *synonymy* - typical NLP problem

Vector Space Model

Documents

Lost glamor

Rated 5 by travel-gini on Feb 26, 2013

Not what it was, but still a place to go.

High tea at Raffles!

Rated 5 by travel-gini on Feb 26, 2013

Amazing service

Great location with a little bit of history, the staff make this hotel though

Have a drink in the Long Bar, throw your nutshells on the floor, then go to the Tiffin Room for the best curry in the world. About £40a head for food but the choice is brilliant and when my wife mentioned it was her birthday at the end of the meal a cake was presented, what amazing service.

Stayed February 2013



Term Document Matrix

	amazing	service	lost	glamour	disappoint	brilliant	super	expensive	noisy	...
Doc1	1	1	0	0	0	1	0	0	0	
Doc2	0	0	1	1	1	0	0	1	0	
Doc3	0	0	0	1	0	0	1	0	0	
Doc4	0	0	0	0	2	0	0	1	1	
...										

Vector Space Model

- Keyword-based, “bag-of-words” approach
- Each item (***document***) is represented by a vector in a n-dimensional space
- Each dimension corresponds to a ***term*** from the vocabulary of a given document collection (***corpus***)
- Documents are pre-processed by operations like *tokenization, case lowering, stop-words removal, stemming*
- Usually using ***TF-IDF*** (Term Frequency - Inverse Document Frequency) *weighting*

TF-IDF Weighting

- To modify the frequency of a word in a document by the perceived importance of the word (the *inverse document frequency*), widely used in information retrieval
 - When a word appears in many documents, it's considered unimportant.
 - When the word is relatively unique and appears in few documents, it's important.

$$tf-idf_{t,d} = tf_{t,d} * idf_t \qquad idf_t = \log \frac{N}{df_t}$$

- $tf_{t,d}$: term frequency – number of occurrences of term t in document d
- idf_t : inverted document frequency of term t
- N : the total number of documents in the corpus
- df_t : the document frequency of term t , i.e., the number of documents that contain the term.

tf-idf Example

TERM VECTOR MODEL BASED ON $w_i = tf_i * IDF_i$											
Query, Q: "gold silver truck" D ₁ : "Shipment of gold damaged in a fire" D ₂ : "Delivery of silver arrived in a silver truck" D ₃ : "Shipment of gold arrived in a truck" D = 3; IDF = log(D/df _i)											
		Counts, tf_i						Weights, $w_i = tf_i * IDF_i$			
Terms	Q	D ₁	D ₂	D ₃	df _i	D/df _i	IDF _i	Q	D ₁	D ₂	D ₃
a	0	1	1	1	3	3/3 = 1	0	0	0	0	0
arrived	0	0	1	1	2	3/2 = 1.5	0.1761	0	0	0.1761	0.1761
damaged	0	1	0	0	1	3/1 = 3	0.4771	0	0.4771	0	0
delivery	0	0	1	0	1	3/1 = 3	0.4771	0	0	0.4771	0
fire	0	1	0	0	1	3/1 = 3	0.4771	0	0.4771	0	0
gold	1	1	0	1	2	3/2 = 1.5	0.1761	0.1761	0.1761	0	0.1761
in	0	1	1	1	3	3/3 = 1	0	0	0	0	0
of	0	1	1	1	3	3/3 = 1	0	0	0	0	0
silver	1	0	2	0	1	3/1 = 3	0.4771	0.4771	0	0.9542	0
shipment	0	1	0	1	2	3/2 = 1.5	0.1761	0	0.1761	0	0.1761
truck	1	0	1	1	2	3/2 = 1.5	0.1761	0.1761	0	0.1761	0.1761

Note that in this example, stopwords and very common words are not removed, and terms are not reduced to root terms.

<http://www.miislita.com/term-vector/term-vector-3.html>

Cosine Similarity

- A similarity measure between two vectors by measuring the cosine of the angle between them

$$Sim(D_i, D_j) = \frac{D_i \bullet D_j}{|D_i| * |D_j|} = \frac{\sum_k w_{ki} w_{kj}}{\sqrt{\sum_k w_{ki}^2 \sum_k w_{kj}^2}}$$

- Example: Given 3 document vectors shown here

$$|D_1| = \sqrt{0.1761^2 + 0.4771^2 + 0.1761^2} = \sqrt{0.2896} = 0.5382$$

$$|D_2| = \sqrt{0.4771^2 + 0.4771^2 + 0.1761^2 + 0.1761^2} = \sqrt{0.5173} = 0.7192$$

$$|D_3| = \sqrt{0.1761^2 + 0.4771^2 + 0.9542^2 + 0.1761^2} = \sqrt{1.2001} = 1.0955$$

$$Sim(D_1, D_2) = (0.1761 * 0.1761) / (0.5382 * 0.7192) = 0.0801$$

$$Sim(D_1, D_3) = (0.4771 * 0.9542 + 0.1761 * 0.1761) / (0.5382 * 1.0955) = 0.8246$$

D ₁	D ₂	D ₃
0	0	0
0	0	0.1761
0	0.4771	0
0	0	0.4771
0	0.4771	0
0.1761	0.1761	0
0	0	0
0	0	0
0.4771	0	0.9542
0	0.1761	0
0.1761	0	0.1761

Example Application: Web Recommenders

- Tracking the user's browsing behaviour to recommend web pages
- Personalized model – keywords related to the user's interests
- **Explicit** preferences
 - user providing positive and negative feedback on pages
- **Inferred** preferences
 - from actions like bookmarking a page
 - From web pages the user visits, and pages one link away from them
- What about forgetting?
 - Short-term vs long-term profile – e.g. short-term model based on tf-idf, long-term model based on a naïve Bayesian classifier
 - Temporal decay of interests


Example Application: Movie Recommenders

- Content-based movie recommender systems.
 - Usually text categorization to learn a user model from the synopses of movies rated by the user (but not restricted to such methods only)
 - User to rate a minimum number of movies into categories such as *terrible, bad, below average, above average, good, and excellent*
 - E.g. INTIMATE, Movies2GO



The titles listed in the 'More like this' section are generated from a variety of information, including genres, country of origin, actors, and much more. Different devices may give slightly different options in the 'More like this' section, to offer a wider range of suggestions for what to watch next.

Example Application: Music Recommenders

- Content-based music recommendation systems
 - Pandora Internet Radio
 - Music Genome Project database: manual content-based description (>400 musical attributes) about qualities of melody, harmony, rhythm, form, composition and lyrics
 - Annotated by experts in music theory (hard to scale)
 - User feedback using thumbs-up or thumbs-down, add music to a station (as positive example)
 - FOAFing the Music
 - Content-based descriptions extracted from music related RSS feeds and the audio itself

Advantages Against CF Approach

	Content-based	Collaborative Filtering
User independence	Just needs the active user's information to build his profile	Requires ratings from other users to find users with similar tastes (rated same items similarly)
Transparency	Recommendations can be explained by listing content features that caused an item to appear in the recommendation list.	Black box (unknown users with similar tastes like that item)
New item	Can recommend items not yet rated by any user.	new item has to be rated by sufficient number of users to get recommended.

Disadvantages

- **Limited content analysis**
 - Limited number and type of features; not sufficient to differentiate what user likes from what he does not like.
 - May need domain knowledge (ontologies)
- **New User**
 - Enough ratings/interactions needed to form user profile for reliable recommendations
- **Over-specialization**
 - Recommended items are *similar* to items that user rated high.
 - *Serendipity* problem: rarely anything *novel* recommended

Semantic Representation of Content

- To overcome the issue of string-matching keyword-based representation
- Bring **semantics** into recommendation
 - Using knowledge sources (e.g. lexicon, taxonomy, ontology, etc)
 - To move from vectors of keywords to vectors of concepts, synsets (WordNet), categories/classes
- Common-sense and domain-specific knowledge may help, too.
 - Augmenting text representation with natural concepts derived from Wikipedia
 - Still in research
- Or, explore innovative representation using **word embeddings**!

Novel Representation of Content

- With user generated content like *folksonomy*, a taxonomy generated by users who collaboratively annotate and categorize resources of interests with freely chosen keywords called *tags*



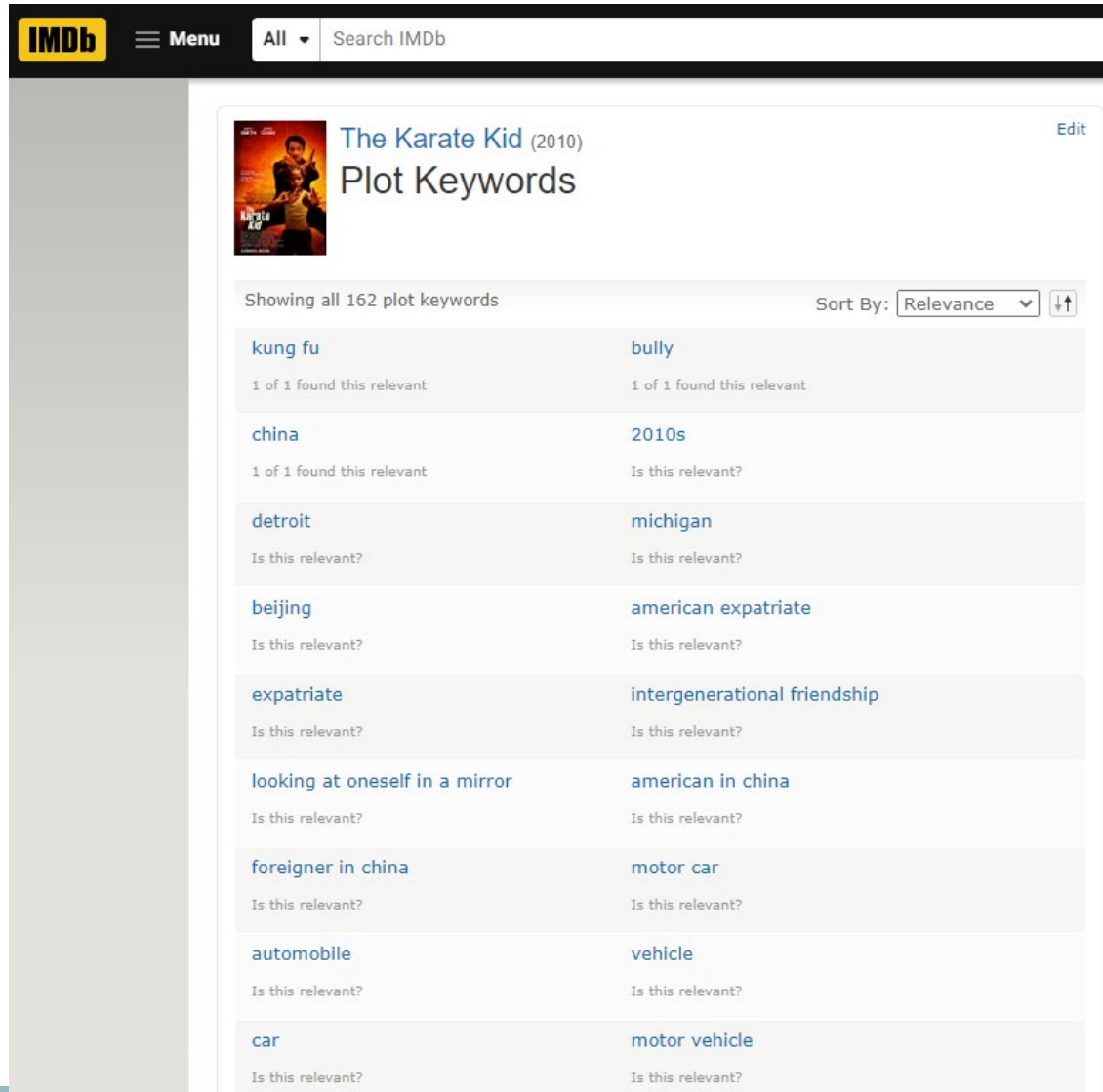
Tag-Based Representation

- Using tags to represent content instead of keywords extracted from descriptions
- E.g. tag-based movie recommendation
- Issues
 - Polysemy and synonymy of tags
 - Users with different expertise and purposes
 - resulting in tags with various levels of abstraction to describe a resource
 - Chaotic proliferation of tags



Movies: Plot Keywords (Tags)


- *IMDb: A keyword is a word (or group of connected words) attached to a title (movie / TV series / TV episode) to describe any notable **object**, **concept**, **style** or **action** that takes place during a title. The main purpose of keywords is to allow visitors to easily search and discover titles.*
- *Contributed by users.*



The screenshot shows the IMDb page for the movie "The Karate Kid (2010)". The page title is "The Karate Kid (2010) Plot Keywords". It indicates that there are 162 plot keywords in total. The keywords are displayed in a grid, sorted by relevance. Each keyword is a blue link, and below it is a small text indicating how many users found it relevant (e.g., "1 of 1 found this relevant").

Keyword	Relevance Count
kung fu	1 of 1 found this relevant
bully	1 of 1 found this relevant
china	1 of 1 found this relevant
2010s	Is this relevant?
detroit	Is this relevant?
michigan	Is this relevant?
beijing	Is this relevant?
american expatriate	Is this relevant?
expatriate	Is this relevant?
intergenerational friendship	Is this relevant?
looking at oneself in a mirror	Is this relevant?
american in china	Is this relevant?
foreigner in china	Is this relevant?
motor car	Is this relevant?
automobile	Is this relevant?
vehicle	Is this relevant?
car	Is this relevant?
motor vehicle	Is this relevant?

IMDb: Plot Keyword-based Browsing



Menu

All ▾ Search IMDb

IMDbPro

+ Wa

Explore Keywords

Enter a keyword:

Browse Interesting Keywords

action hero
alternate history
ambiguous ending
americana
anime
anti hero
avant-garde
b movie
bank heist
based on book
based on play
based on comic
based on comic book
based on novel
based on novella
based on short story
battle
betrayal
biker
black comedy
blockbuster
bollywood
breaking the fourth wall
business
caper
car accident
car chase
car crash

kidnapping
knight
kung fu
macguffin
medieval times
mockumentary
monster
mother daughter relationship
mother son relationship
multiple actors playing same role
multiple endings
multiple perspectives
multiple storyline
multiple time frames
murder
musical number
neo noir
neorealism
ninja
no background score
no music
no opening credits
no title at beginning
nonlinear timeline
on the run
one against many
one man army
opening action scene

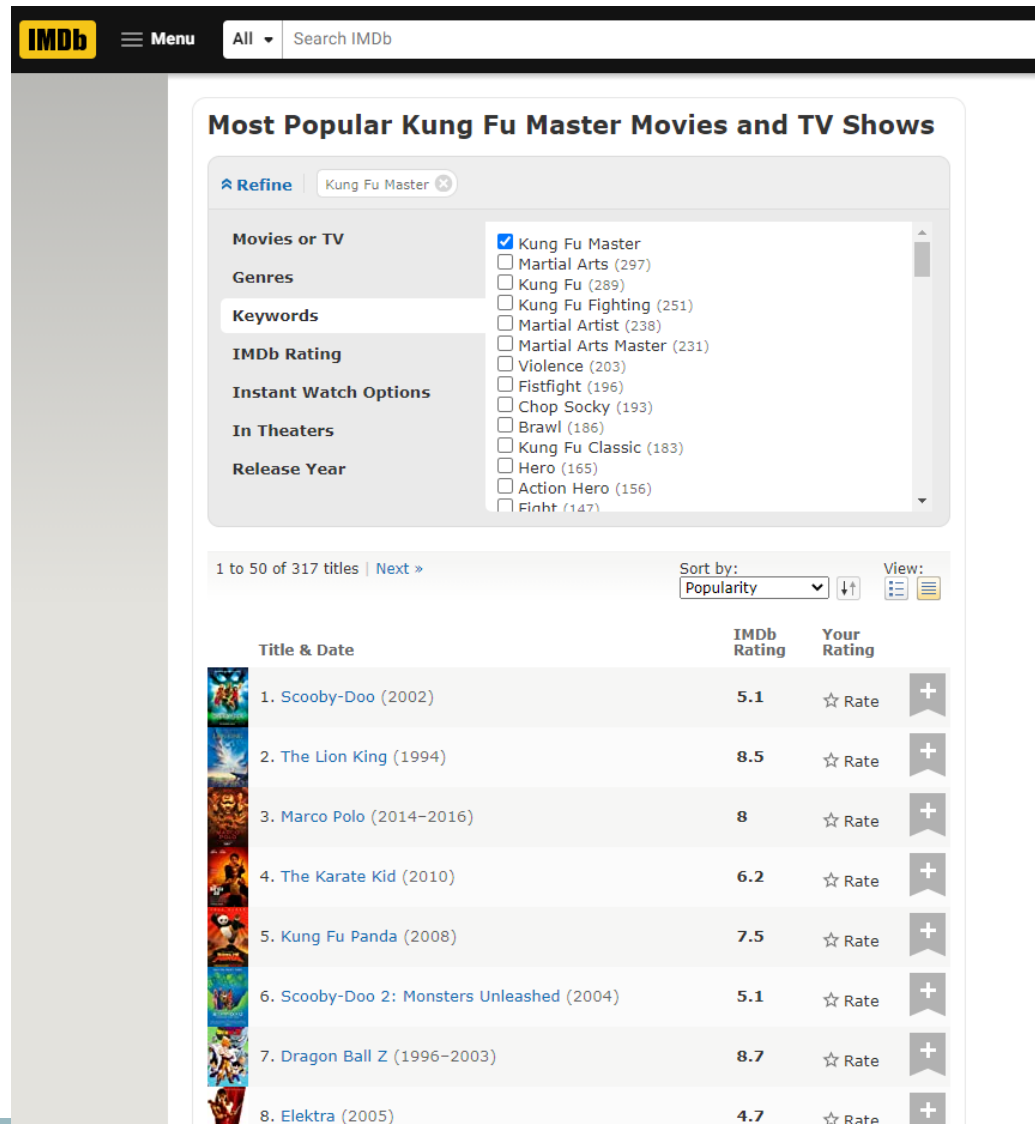
Browse Titles By:

Genre
Country
Language
Year

Advanced Search:

Advanced Title Search
Advanced Name Search
Collaborations and Overlaps

IMDb: Plot Keyword-based Browsing



IMDb Menu All Search IMDb

Most Popular Kung Fu Master Movies and TV Shows

[Refine](#) Kung Fu Master

- Movies or TV**
- Genres**
- Keywords**
- IMDb Rating**
- Instant Watch Options**
- In Theaters**
- Release Year**

- ☒ Kung Fu Master
- ☐ Martial Arts (297)
- ☐ Kung Fu (289)
- ☐ Kung Fu Fighting (251)
- ☐ Martial Artist (238)
- ☐ Martial Arts Master (231)
- ☐ Violence (203)
- ☐ Fistfight (196)
- ☐ Chop Socky (193)
- ☐ Brawl (186)
- ☐ Kung Fu Classic (183)
- ☐ Hero (165)
- ☐ Action Hero (156)
- ☐ Fight (147)

1 to 50 of 317 titles | [Next »](#)

Sort by: Popularity View: [Icons]

Title & Date	IMDb Rating	Your Rating
1. Scooby-Doo (2002)	5.1	☆ Rate +
2. The Lion King (1994)	8.5	☆ Rate +
3. Marco Polo (2014–2016)	8	☆ Rate +
4. The Karate Kid (2010)	6.2	☆ Rate +
5. Kung Fu Panda (2008)	7.5	☆ Rate +
6. Scooby-Doo 2: Monsters Unleashed (2004)	5.1	☆ Rate +
7. Dragon Ball Z (1996–2003)	8.7	☆ Rate +
8. Elektra (2005)	4.7	☆ Rate +

Social Tags

- To relieve the new user problem
- Tags from items rated by the user
- Tags adopted by other users who rated the same items (*social tags*)
- Include social tags in the user profile
 - ➔ hybrid approach combining content-based and collaborative approaches



Serendipity

- To overcome the problem of over-specialization, and achieve *diversity* of recommendations
- Relevant, and novel/unexpected

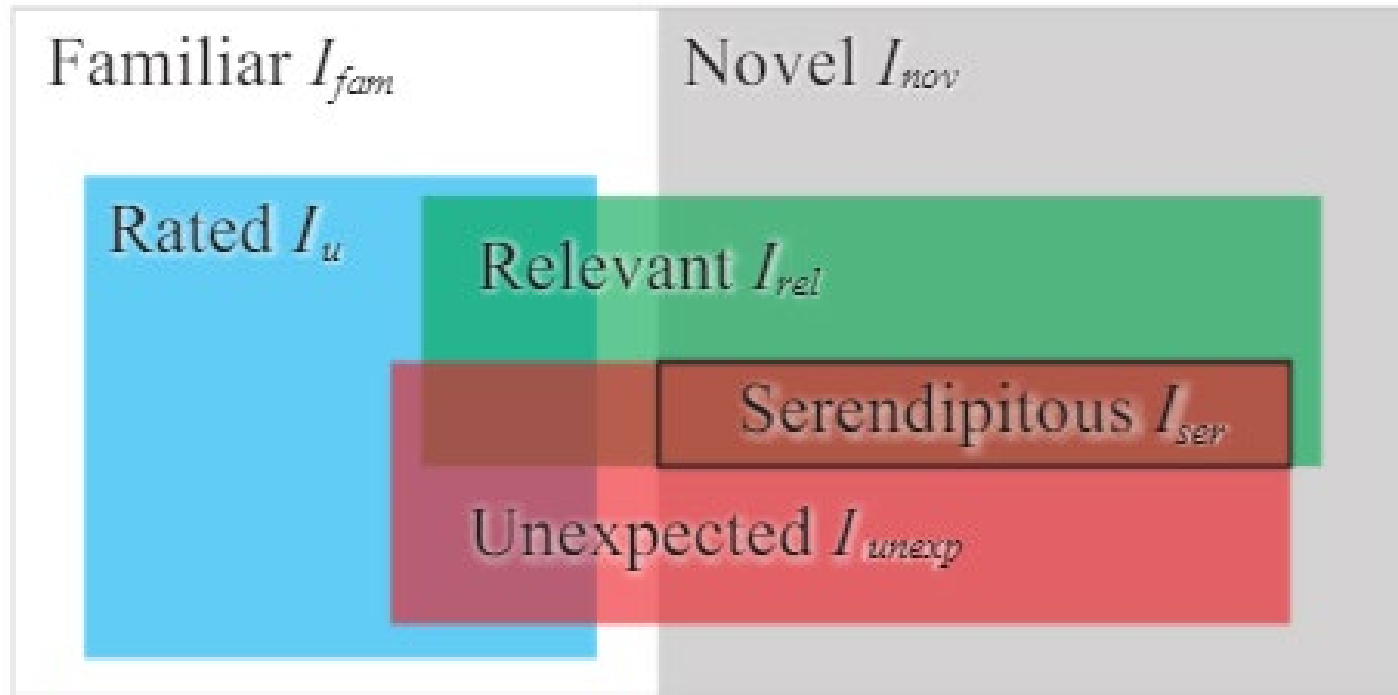
serendipity

noun [U] • formal

UK  /ˌser.ənˈdɪp.ə.ti/ US  /ˌser.ənˈdɪp.ə.ti/

the fact of finding interesting or valuable things by chance

The Concept of Serendipity



Kotkov, Denis, Shuaiqiang Wang, and Jari Veijalainen. "A survey of serendipity in recommender systems." *Knowledge-Based Systems* 111 (2016): 180-192.

Ways to Serendipity

- Introduce randomness
 - Completely random item
 - Bounded random selection: from K best matches
- Avoid recommending items that are too similar to what the user has seen, i.e. filter off those above a similarity threshold
- How to evaluate? User-centric methods probably.

