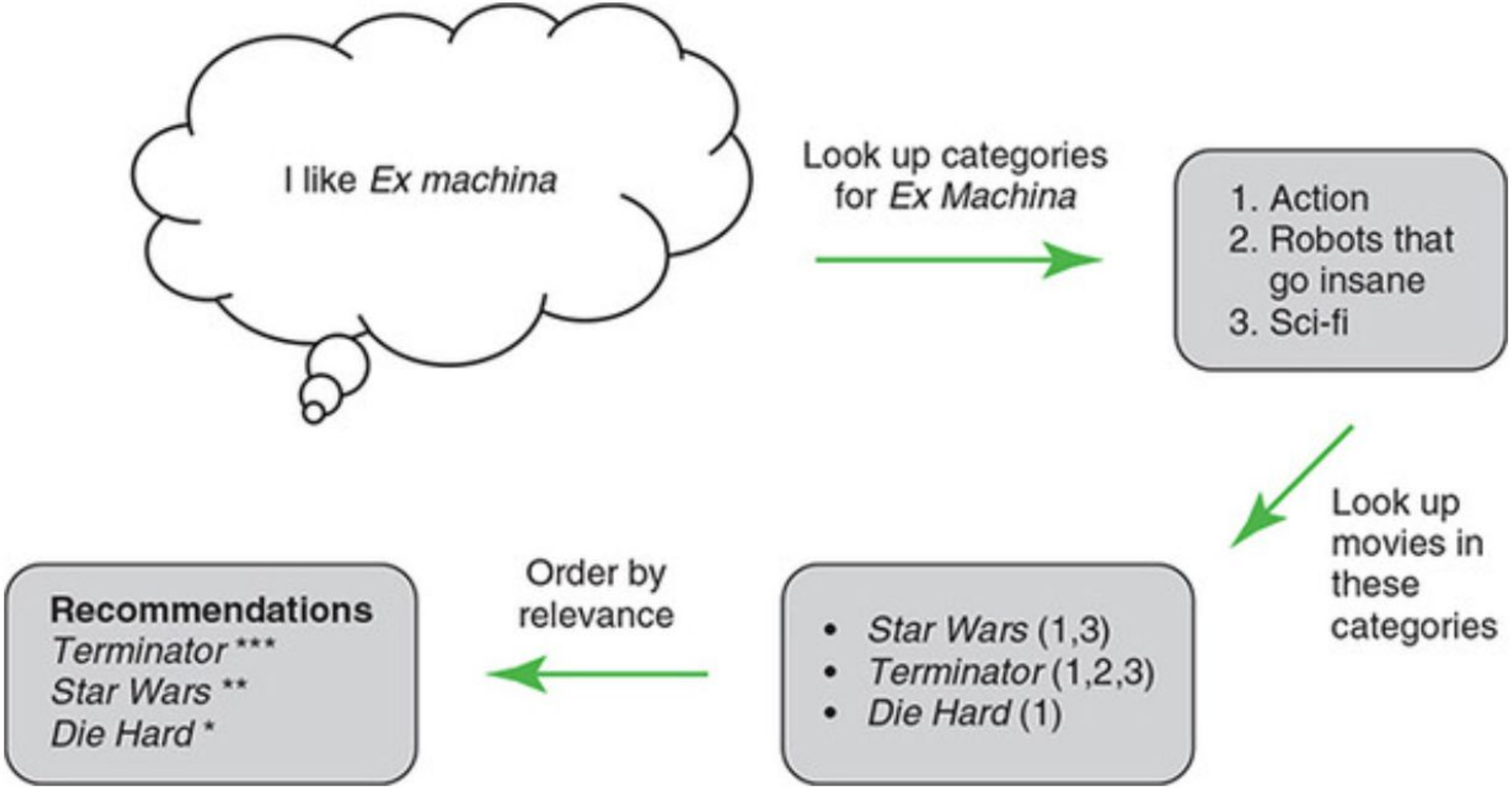
Content Filtering

What about metadata you know about content and the user? Is a recommender good if it doesn’t take those things into account?

- *Me:* I just saw *Ex Machina* (okay, still haven’t watched it but I look forward to it).
- *Imaginary interested person:* Really, was it good?
- *Me:* Yeah, there were some very interesting subjects (imagining that I watched it).
- *Imaginary interested person:* All right, so you like robot people.
- *Me:* Well, yes (feeling like I shouldn’t say yes).
- *Imaginary interested person:* Technology that goes bad. Then you must like *Terminator*.
- *Me:* Yes (relieved).

“More Like These” Recommendations



I like *Ex machina*

Look up categories for *Ex Machina*

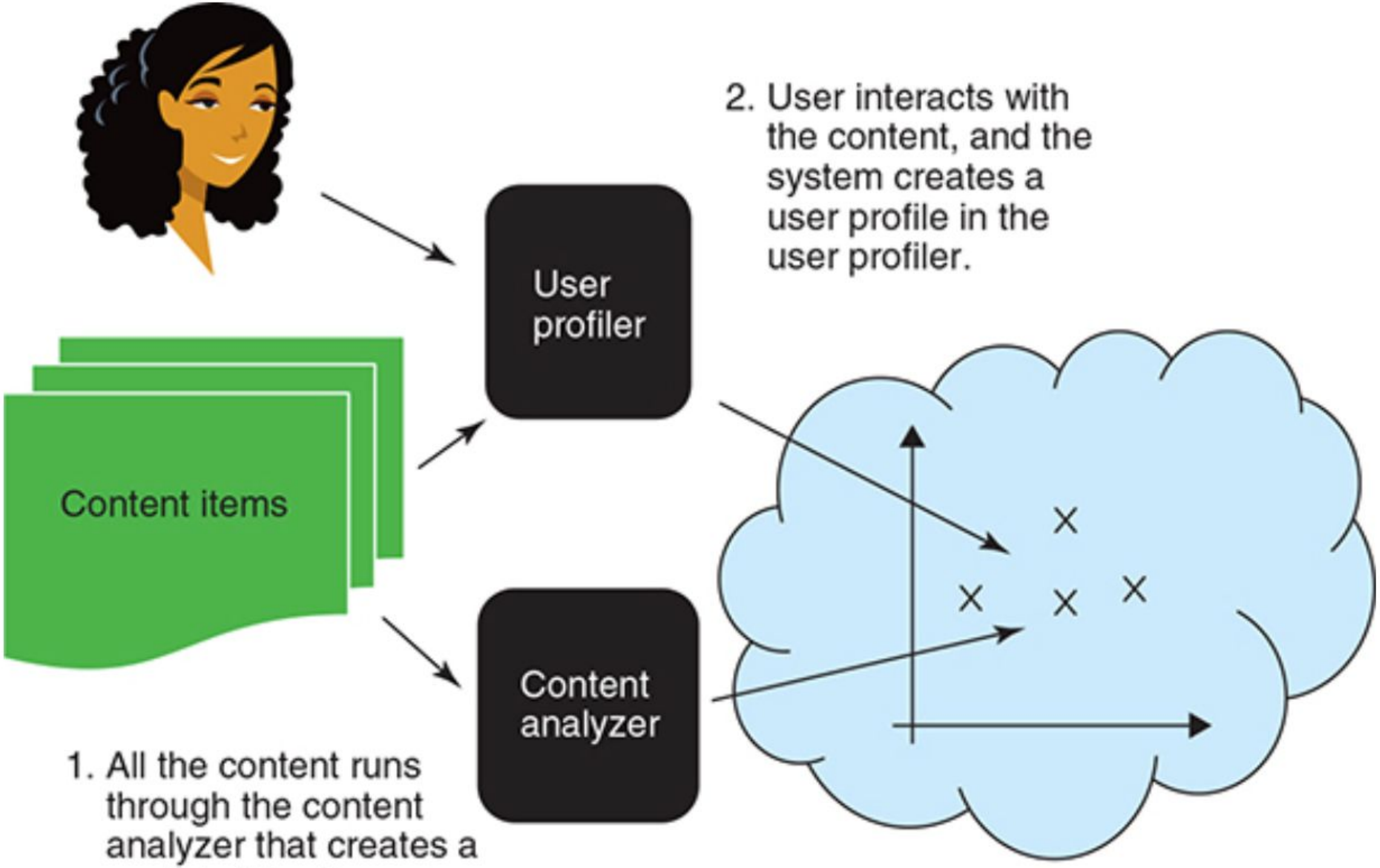
1. Action
2. Robots that go insane
3. Sci-fi

Look up movies in these categories

- *Star Wars* (1,3)
- *Terminator* (1,2,3)
- *Die Hard* (1)

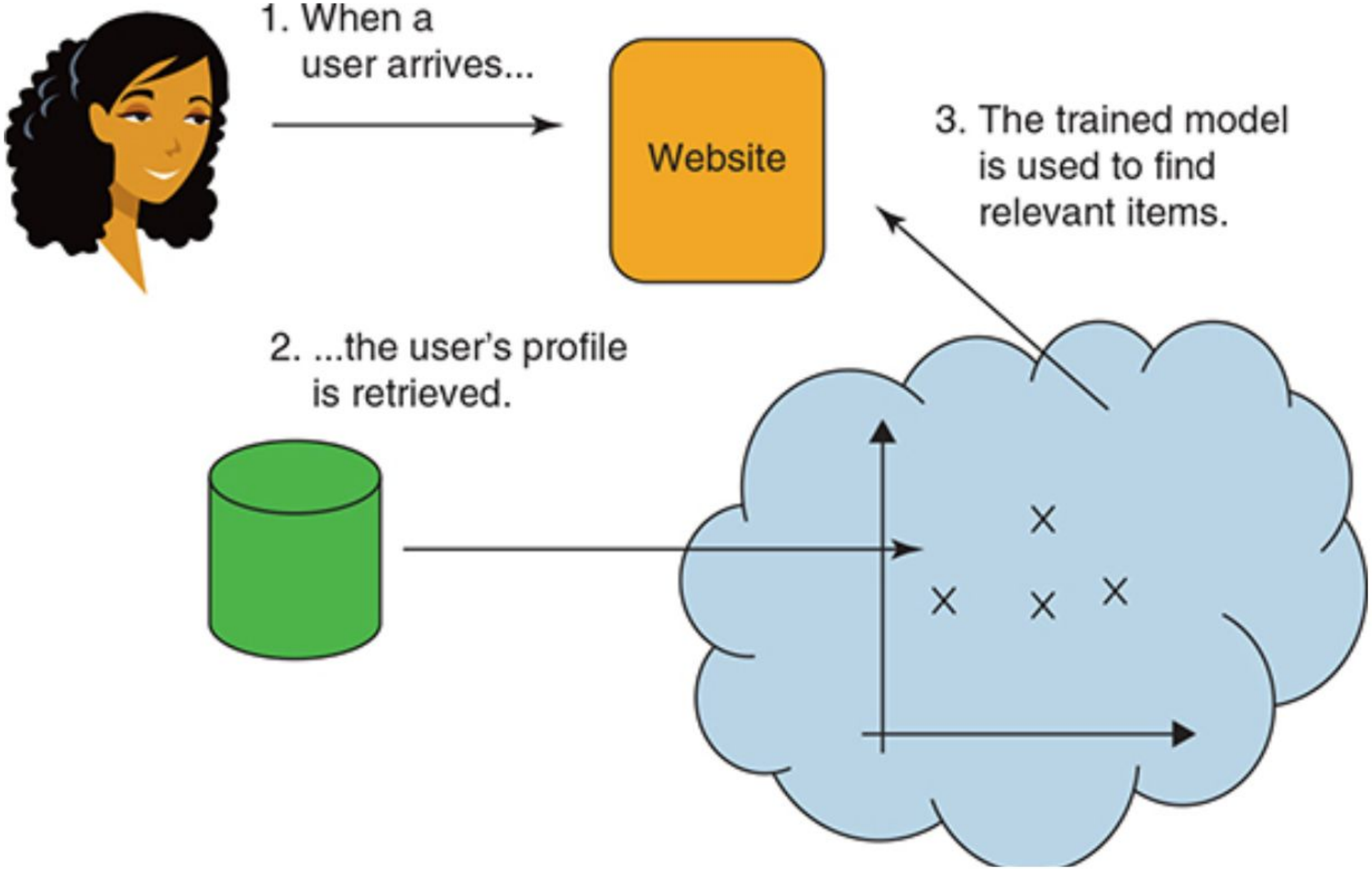
Order by relevance

Recommendations
Terminator ***
Star Wars **
Die Hard *



2. User interacts with the content, and the system creates a user profile in the user profiler.

1. All the content runs through the content analyzer that creates a model of the items.



1. When a user arrives...

2. ...the user's profile is retrieved.

3. The trained model is used to find relevant items.

Three things needed for Implementation

- Content analyzer
- User profiler - It can be a simple list of items consumed by the user
- Item retriever - Retrieves relevant items by comparing the user profiles to the content profiles

Content Analyzer

Metadata of the content is used

Extract features from the metadata

Extracting features from descriptions

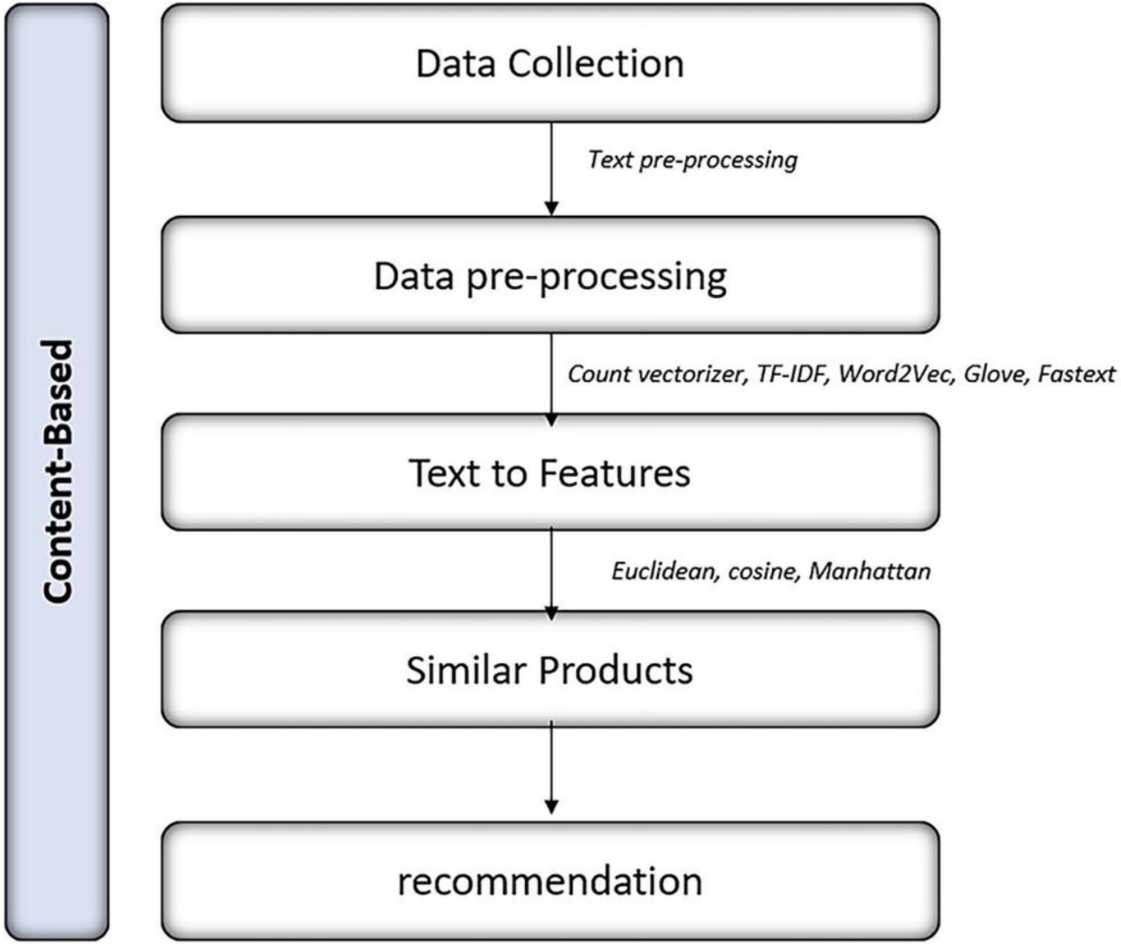
Algorithms like TF-IDF and LDA or NLP can be used to extract features from text

User Profile

- Create a user profile that encompasses all the items or content the user likes
- Iterate through each item in the user list and find similar products
- LDA, TF-IDF or NLP techniques can be used to create user profiles

Item Retriever

- Calculate the similarity between user profile and the item
- Retrieve similar items
- Use business rules to filter and order
- Serve recommendations


Content-Based
Data Collection
Text pre-processing
Data pre-processing
Count vectorizer, TF-IDF, Word2Vec, Glove, Fastext
Text to Features
Euclidean, cosine, Manhattan
Similar Products
recommendation

Pros

- New items are easy to add. Create the item feature vector, and you’re set to go.
- You don’t require much traffic. Because you can find similarity based on content descriptions, you can start recommending things from the first visit or rating.
- It recommends across popularity; content-based recommenders don’t care which content is popular right now if it finds that a film nobody ever watched is as likely to be recommended as one that everybody watched.

Cons

- Conflates liking with importance. If you like science fiction films with Harrison Ford, the system will also give you films with Harrison Ford that aren’t science fiction.
- No serendipity; it’s specialized.
- Limited understanding of content. It might be hard to include all features that mark the aspects that make content favorable to a user, which means that the system can easily *misunderstand* what the user likes.