

A graphic of a movie film strip with sprocket holes. The text "A Non-intrusive Movie Recommendation System" is centered on a black background within the strip. The film strip has orange markings at the top and bottom, including "019HJ4 00 RVP" and "00" and "00A".

A Non-intrusive Movie Recommendation System

Tania Farinella¹, Sonia Bergamaschi², and Laura Po²

¹ **vfree.tv** GmbH, München, Germany

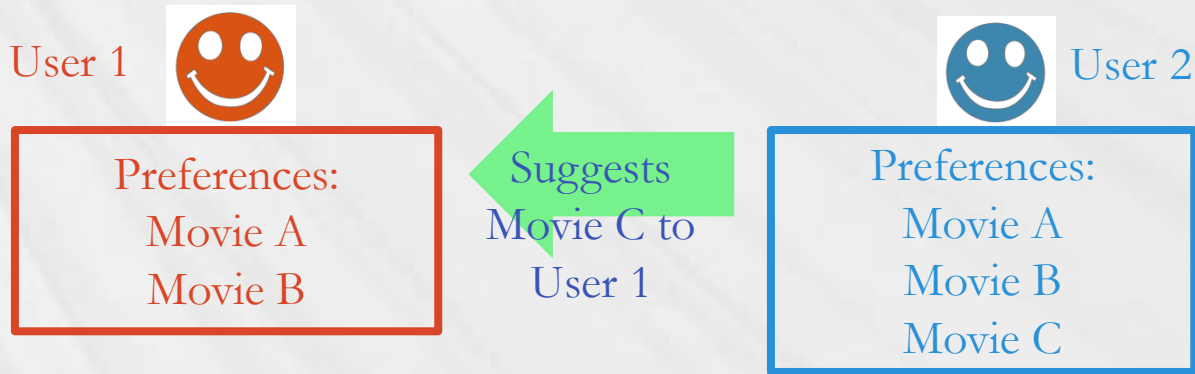
² **Department of Engineering “Enzo Ferrari”**,
University of Modena and Reggio Emilia, Italy

Recommendation Systems

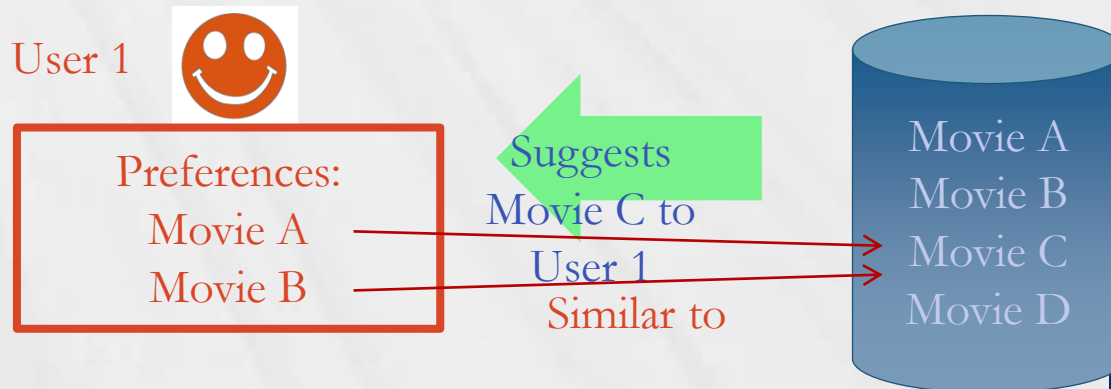
- ❖ **Recommendation systems** are information filtering systems that recommend products available in e-shops, entertainment items (books music, videos, Video on Demand, books, news, images, events etc.) or people (e.g. on dating sites) that are likely to be of interest to the user
- ❖ Recommendation algorithms are the basis of the targeted advertisements that account for most commercial sites' revenues
- ❖ The DVD rental site Netflix deemed its recommendation algorithms such important that it offered a million-dollar prize in 2009 to anyone who could improve their predictions by 10%.

Collaborative filtering vs Content-based

- ❖ **Collaborative filtering** makes automatic predictions (filtering) of the interests of a user by collecting preferences or taste information from many users (collaborating)



- ❖ **Content-based approach** recommends items based on a comparison between their content and a user profile



Movie Recommendation Systems

- ❖ Among Recommendation Systems, the widely utilized **collaborative filtering systems** focus on the analysis of user roles or user ratings of the items.
 - ◆ These systems decrease their performance at the start-up phase and due to privacy issues, when a user hides most of his personal data
- ❖ **Contents-based recommendation systems**, in the context of movies, compare movie features to suggest similar multimedia contents
 - ◆ these systems are based on less invasive observations, however they are less effective in supplying tailored suggestions

Movie Recommendation systems

- ❖ There are dozens of Movie Recommendation engines on the Web. Some require little or no input before they give you Movie titles, while others want to find out exactly what your interests are



Netflix asks you to rate movies to determine which films you'll want to see next



Rotten Tomatoes requires you to say what kind of films you enjoy, which actors you want to see, and other criteria to help it in finding the best movie for you




Movielens evaluates your tastes based on ratings to films you've seen before. Once you rate at least 15 movies, it returns recommendations



IMDb automatically recommends films similar to the movie you search for



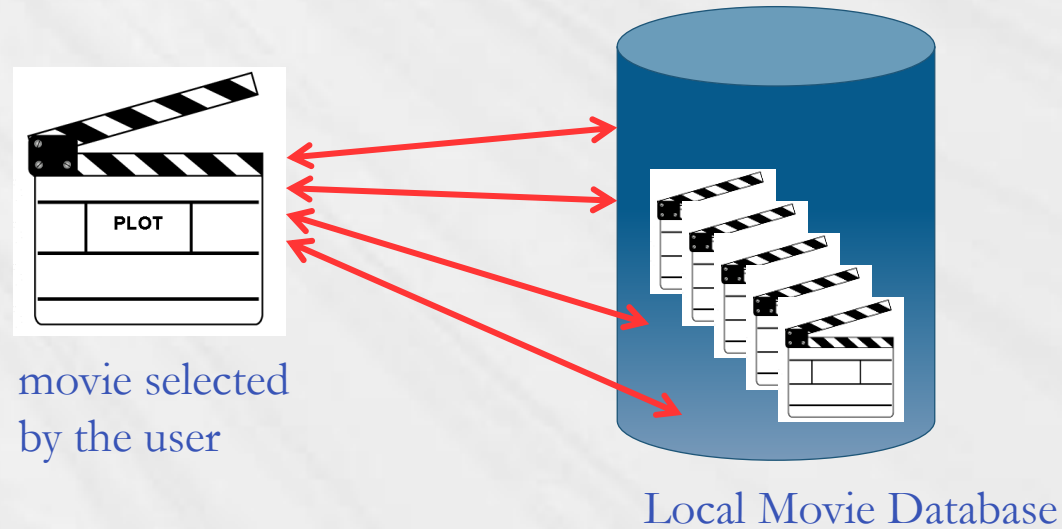
Jinni is able to find films based on your mood, time available, setting, or reviews



Goal -> a Plot-based Movie RS

- ❖ Our goal was to develop a **content-based** Movie Recommendation System **based on the computation of plots similarity**
- ❖ Our **Plot-based Recommendation System** evaluates the similarity among the plot of a video that was watched by the user and a large amount of plots stored in a Local Movie Database
- ❖ Since it is independent from user ratings, it is able to propose famous and beloved movies as well as old or unheard movies or programs strongly related to the contents of the video the user has watched

Plot-based Movie RS

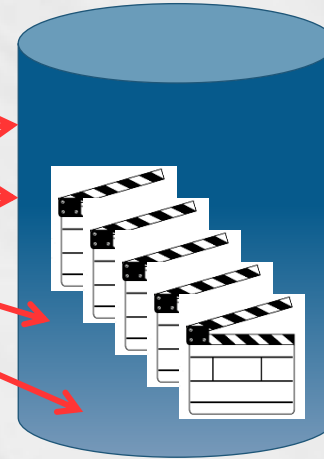


- ❖ Movies can be compared considering only plots

Plot-based Movie RS



movie selected
by the user



Local Movie Database

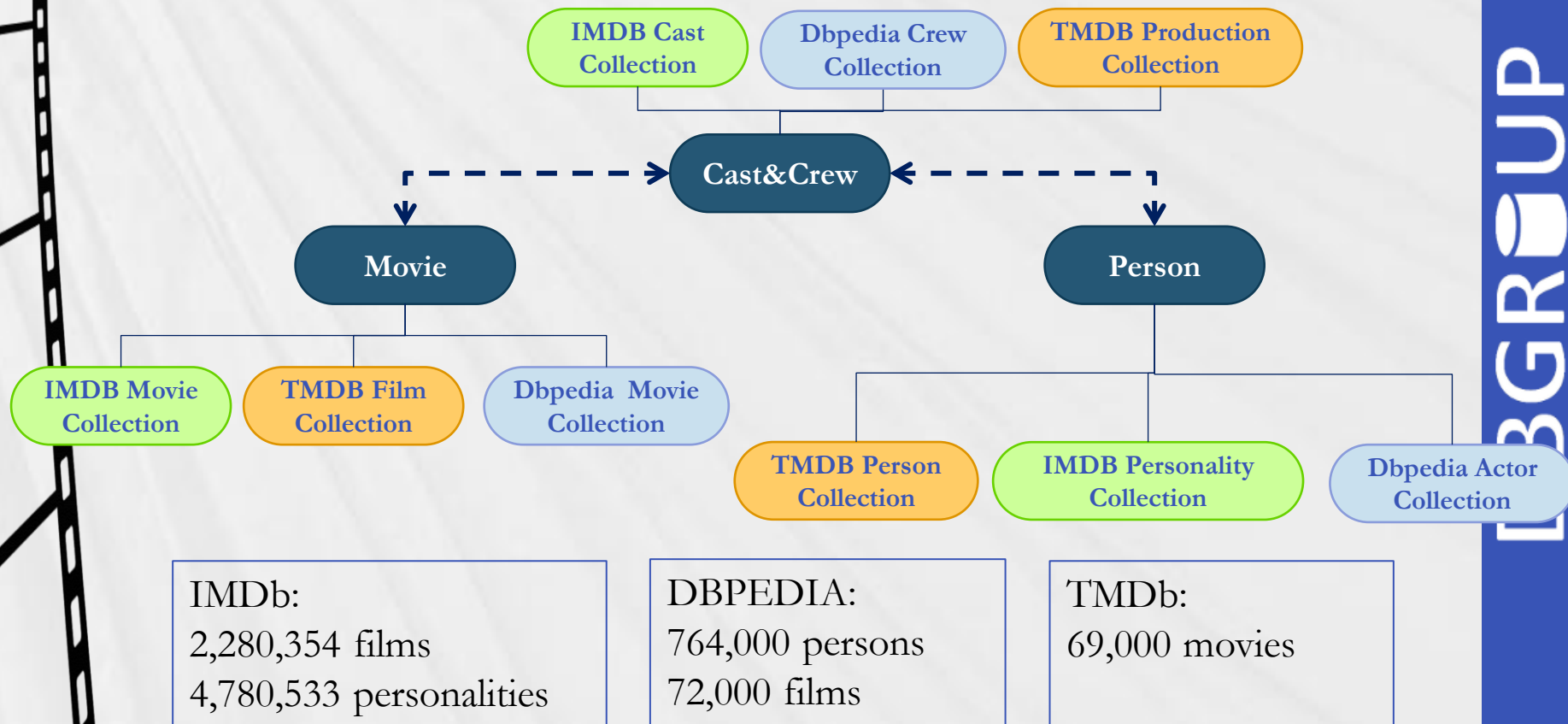
- ❖ The similarity of the plots can be combined with the similarity of other features such as directors, genre, producers, release year, cast etc



The Local Movie Database

- ❖ Our first goal was to generate an extensible and reliable collection of multimedia metadata
- ❖ We studied the major movie repositories:
 - ◆ **Internet Movie Database (IMDb)**
 - ◆ **Dbpedia**
 - ◆ **Open Movie Database (TMDb)**
- ❖ The Local Movie Database integrates the above databases
- ❖ The choice of **MongoDB** as local database server
 - ◆ It is a scalable, high-performance, open source **NoSQL** database server
 - ◆ **Document-oriented storage** - It does not require a fixed schema, different collections in the databases may have different structure and store different attributes. This makes the integration of data easier and faster

The Local Movie database



Plots Modelling: Vector space model

- ❖ Each plot is represented as a vector of keywords with associated weights
- ❖ Weights depend on the distribution of keywords in the given training set of plots

- ❖ **Matrix T**

The weight of keyword 2 according to plot b

	keyword1	keyword2	...
plot a			
plot b		$w_{b,2}$	
plot c			

- ❖ The **cosine similarity** is used as a distance metric to calculate the similarity score between two texts – it is used to either compare descriptions within the training set or descriptions that are not included in the training set

Vectors that represent two different plots

$$\cosin(v_i, v_j) = \frac{\sum_k (v_i[k] \cdot v_j[k])}{\sqrt{\sum_k v_i[k]^2} \cdot \sqrt{\sum_k v_j[k]^2}}$$

Weights Computation

- ❖ **Pre-processing:** plots need to be converted into vectors of keywords
 - ◆ stop words removal
 - ◆ lemmatization by TreeTagger
- ❖ **1st step:** weights are defined as occurrences of keywords in the description
- ❖ **2nd step:** weights are modified by **tf-idf** or **log** techniques
- ❖ **3th step:** the **matrix T** is transformed by performing **Latent Semantic Analysis**

Weighting techniques

- ❖ A weight represents the relevance of a specific keyword according to a specific text
- ❖ *Term Frequency-Inverse Document Frequency (tf-idf)* technique

frequency of the keyword k
in the vector v

$$weight_{tf-idf} = \frac{tf(k, d) \cdot idf(k)}{\sqrt{\sum_j v[j]^2}}$$

depends on the number N of
vector descriptions in the corpus
and on the number of vector
descriptions in which the keyword k
appears

- ❖ *Log Entropy (log)* technique

$$weight_{log} = weight_{local} \cdot weight_{global}$$

$$weight_{local} = \log(f_{i,j} + 1)$$

frequency of the keyword (k_j)
in the vector description (d_i)

number of vector
descriptions within
the corpus

$$weight_{global} = 1 + \frac{\sum_{l=1}^N P(d_l, k_j) \cdot \log(P(d_l, k_j))}{\log(N)}$$

LSA (Latent Semantic Analysis)

- ❖ The **LSA** is a technique for analyzing relationships between a set of documents and the terms they contain by producing a set of concepts related to the documents and terms. LSA assumes that the words that are close in meaning will occur in similar pieces of text.
- ❖ To generate small and non-sparse vectors (about 500 elements versus 2000000 keywords) of matrix T obtained by the cited weighting techniques we applied a mathematical technique called singular value decomposition (SVD)
- ❖ Each row and column of the resulting **matrix T'** can be represented as a vector combination of the eigenvectors of the matrix $T' T'^T$

$$v = \sum_i \text{coefficient}_i \cdot \text{eigenvector}_i$$

- ❖ Where the coefficients of the above formula represent how strong the relationship between a keyword (or a description) and a topic eigenvector is. The eigenvectors define the so-called **topic space**, thus, the coefficients related to a vector v represent a topic vector.

LSA - topic vectors

- ❖ The topic representation of the keywords is used as a natural language model in order to compare texts
- ❖ Topic vectors are linear combinations of keywords, they may be useful for three main reasons:
 - ◆ (1) as the number of topics that is equal to the number of non-zero eigenvectors is usually significantly lower than the number of keywords, the topic representation of the descriptions is more compact;
 - ◆ (2) the topic representation of the keywords makes it possible to add movies that have been released after the definition of the matrix T' without re-computing the matrix T' ;
 - ◆ (3) to find similar movies starting from a given one, we just need to compute the topic vectors for the plot of the movie and then compare these vectors with the ones we have stored in the matrix T' finding the top relevant.

TF-IDF vs LSA - an example

There is a **mouse** below the new **Ferrari** that is parked in front of the **market**.

With one **mouse** click you can view all available **cars** and thus renders going to the **shop unnecessary**.

	ferrari	market	mouse	click	car	shop
v1	0.67	0.38	0.48	0	0	0
v2	0	0	0.41	0.54	0.48	0.39
product	0	0	> 0	0	0	0

vectors generated by using the **tf-idf** technique

	ferrari	market	mouse	click	car	shop
v1	0.67	0.38	0.48	0	>0	>0
v2	>0	>0	0.41	0.54	0.48	0.39
product	>0	>0	>0	0	>0	>0

vectors generated by using the **LSA** technique

- ❖ **tf-idf** is not able to recognize either synonyms (e.g. market and shop) or hyponyms (e.g. Ferrari and cars); in contrast, the use of **LSA** emphasizes underlying semantics

Evaluation

- ❖ We have performed several evaluations:
 - ◆ **Manual tests** to evaluate and compare the weighting techniques
 - ◆ An evaluation of the **computational costs** of the model and its approximations
 - ◆ An evaluation of the **user judgments** on the lists of recommended movies from IMDb and LSA

Manual tests

- ❖ 1) Results obtained by applying tf-idf and log techniques on the local database show slight differences, and the quality does not seem to be significantly different
- ❖ 2) A noticeable quality improvement can be achieved by applying the LSA technique. The outcome of Latent Semantic Analysis is superior to other techniques such as tf-idf or log
- ❖ 3) LSA over tf-idf and LSA over log techniques gives comparable outcomes

Computational Costs Evaluation

- ❖ Given a target plot, all the other plots in the database can be ranked according to their similarity in about 12 seconds (tests have been conducted on the data extracted from Dbpedia: 78602 movies with 133369 keywords appearing in the plots)
- ❖ To further decrease similarity time consumption, three LSA models have been built using different assumptions (i.e. decreasing the number of LSA topics)

	Complete model	Approximate model	Fast model
minimum document frequency	1	10	10
minimum vector length	1	20	20
minimum tf-idf weight	0	0.09	0.14
minimum log weight	0	0.09	0.14
minimum lsa weight	0	0.001	0.001
number of lsa topics	500	350	200
matrix size (rows x columns)	78602 x 133369	68038 x 15869	68038 x 15869
Similarity time cost	12 seconds	7 seconds	5 seconds

User judgement

Choose the movies that you would recommend - survey A

Exit this survey

The Godfather (1972)

HOW TO DO THIS SURVEY: Suppose that one of your friends tells you about a movie he saw and you have to suggest him other similar films. Which would you choose?! Select the movies you think are most similar to the film in the question. If you don't know the movie in the question, you can skip to answer the question simply by going to the next page.

2. Movie: The Godfather (1972) - 175 min - Crime | Drama - Director: Francis Ford Coppola - Writers: Mario Puzo (screenplay), Francis Ford Coppola (screenplay) - Stars: Marlon Brando, Al Pacino and James Caan - Plot: The aging patriarch of an organized crime dynasty transfers control of his clandestine empire to his reluctant son.

Carlito's Way (1993) - R 144 min - Crime | Drama | Thriller - 10 November 1993 (USA) - Plot: A Puerto-Rican ex-con, just released from prison, pledges to stay away from drugs and violence despite the pressure around him and lead on to a better life outside of NYC. - Director: Brian De Palma - Writers: Edwin Torres (novels), David Koepp (screenplay) - Stars: Al Pacino, Sean Penn and Penelope Ann Miller

Goodfellas (1990) - R 146 min - Biography | Crime | Drama - Plot: Henry Hill and his friends work their way up through the mob hierarchy. - Director: Martin Scorsese - Writers: Nicholas Pileggi (book), Nicholas Pileggi (screenplay) - Stars: Robert De Niro, Ray Liotta and Joe Pesci

Donnie Brasco (1997) - R 127 min - Biography | Crime | Drama - Plot: An FBI undercover agent infiltrates the mob and finds himself identifying more with the mafia life to the expense of his regular one. - Director: Mike Newell - Writers: Joseph D. Pistone (book), Richard Woodley (book) - Stars: Al Pacino, Johnny Depp and Michael Madsen

A Non-intrusive Movie Recommendation System

Shadow Hours (2000) - 95 min - Drama | Thriller - Director: Isaac H. Eaton -

- ❖ Our goal was to compare our system recommendations with respect to the IMDb recommendations
- ❖ We asked users to judge the recommendations proposed for the 18 most famous movies (from the ranked list of IMDb)
- ❖ For each title we selected 6 recommended movies with our algorithm and 6 recommended movies suggested by IMDb
- ❖ Then, we asked users to select which movies are the most similar to the target one
- ❖ We collected 146 evaluations from 30 users

User judgement - Results

Total number of movies evaluated = 18

4 cases = LSA selects the best recommendations

10 cases = IMDb selects the best recommendations

4 cases = both systems have the same performance

Comments on the results

- ❖ IMDb system advertises famous movies, while LSA proposes even old or unknown movies, thus is very difficult to gain votes for LSA recommendations
- ❖ IMDb uses factors such as user votes, genre, title, keywords, and, most importantly, user recommendations themselves to generate an automatic response, this system requires a great deal of human support
- ❖ By using the LSA techniques we attempted to find fully automatic ways of generating these strictly content-based recommendations
- ❖ While our system does not outperform the commercial approach, it can be used to make recommendations when knowledge of users' preferences is not available.

Conclusion

- ❖ The developed system takes advantage of the NLP techniques and is able to make recommendations without access to user preference information but just comparing natural language descriptions of the plots
- ❖ We evaluated 3 algorithms (tf-idf, log and LSA) and found out that LSA outperformed log and tf-idf algorithms
- ❖ We compared our system to a commercial system (IMDb) that heavily relies on human volunteered effort. While our system does not outperform the commercial approach, it is much less labor intensive and can be ported to any domain where natural language descriptions exist.



Future Work

- ❖ Keywords extraction might benefit from the use of lexical databases as WordNet as they are particularly helpful in dealing with synonyms and polysemous terms
- ❖ In WordNet, words (i.e. lemmas) are organized in groups of synonyms called synsets. Synsets are connected depending on semantic relationships such as hypernymy and hyponymy
- ❖ **Each keyword might be replaced by its meaning** (synset) before the application of the weight techniques by adopting Word Sense Disambiguation techniques (in order to understand which of the synsets better express the meaning of a keyword in a plot)
- ❖ The semantic relationships between synsets can be used for enhancing the keyword meaning by adding its hypernyms and hyponyms