Hasso Plattner Institut

IT Systems Engineering | Universität Potsdam

Modeling Interestingness and Serendipity in Relevance Search

Maximilian Jenders
Information Systems Group
Hasso Plattner Institute

Agenda

- Motivation
- Challenges

- Modeling Serendipity
  - Corpus
  - Word-Frequency-based
  - Topic-based

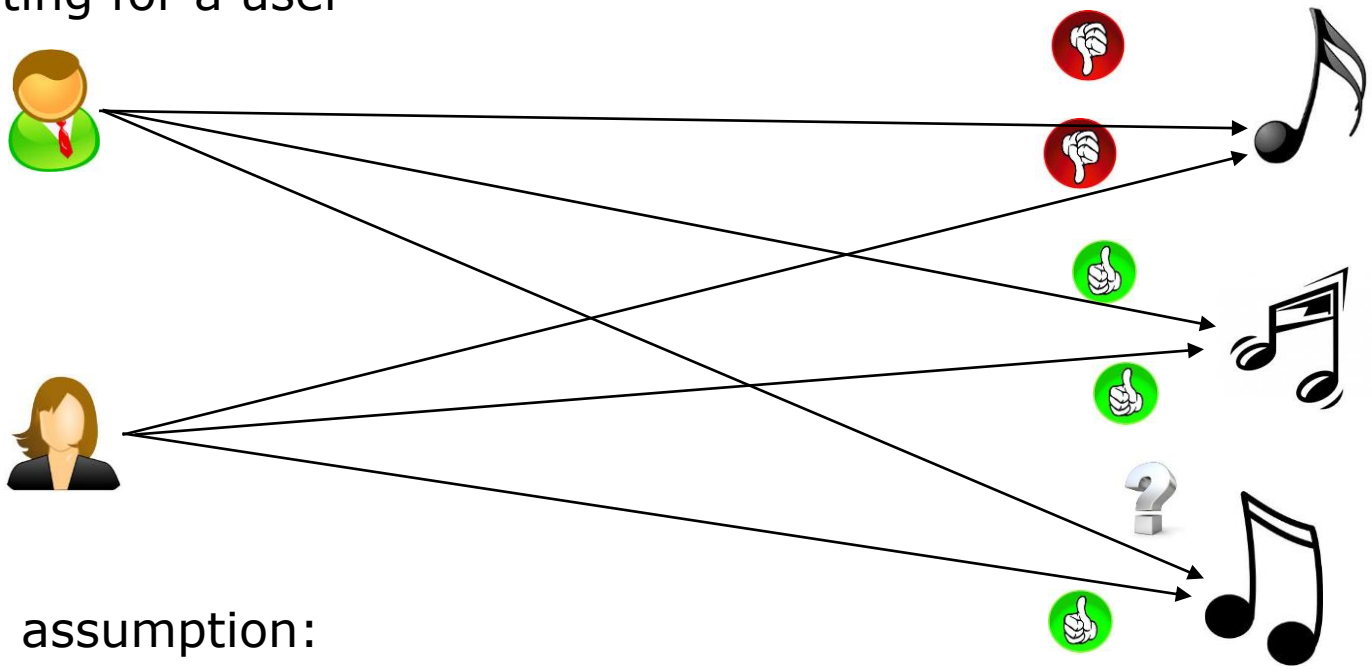
- Shift in research direction

Motivation

- Recommendation Problem
- Focus on text documents
- Serendipity:
  - “Happy Accident”
  - “Pleasant Surprise”
  - “The Three Princes of Serendip”: Princes were always making discoveries by accident of things they were not in quest of

Motivation

- Recommender Systems make predictions about which items are interesting for a user



- Typical assumption:
  - If a user likes an item, he/she will like items with similar features
  - Similar users like similar things

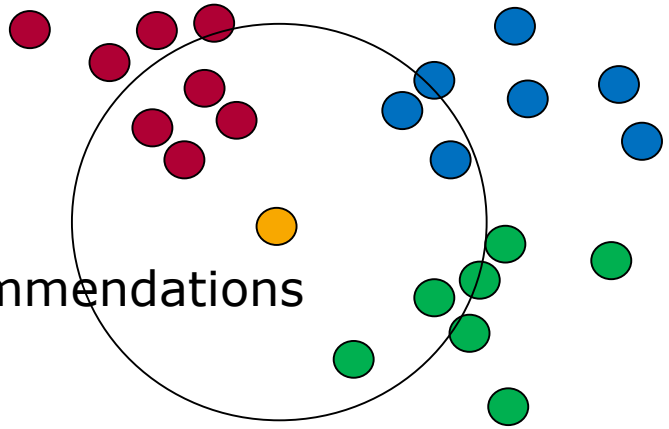
Challenge

- Drawback: Only quantitative analysis
  - User satisfaction might be different for obvious vs. uncertain recommendations

- Focus on similarity measures in combination with accuracy-based metrics may not maximize user experience

Recommendation Enhancements

- Approaches in Related Work:
  - **Diversity**
    - Based on dissimilarity within recommendations
  - **Novelty**
    - Based on user history
  - **Serendipity**
    - Based on reception and expectedness
    - Related Work use traditional recommender system



- Here: focus on text documents, no user information

Agenda

- Motivation
- Challenges

- Modeling Serendipity
  - Corpus
  - Word-Frequency-based
  - Topic-based

- Shift in research direction

Corpus

- NYT corpus:
  - 1 DVD of NYT articles from 1987-2007
  - 1,8 million articles
  - 650.000 manually written article summaries
  - Article text, date, title, author
  - Mentioned Entities (Persons / Locations / Organizations)
- Again: Purely content-based features
- No way to automatically evaluate, need humans to judge interestingness

Models for Serendipity

- Scenario: User reads one article („initial article“), we want to recommend some more articles
- Two models for capturing Serendipity:
- Word-Frequency-Based
  - Which words occur how often in the article
- Topic-Based
  - What topics is the article about

Word-Frequency-Based Model

- Build Word-Frequency-Based Model:
  - Analyze word frequencies of articles
  - Create model from frequencies
  - Build set of similar documents based on frequencies
  - Serendipitous documents share some features with the set, but not all

- Preliminary evaluation with some initial articles from very different topics
  - For some initial articles, recommendations are generally good
  - For others, all recommendations are very bad
  - Overlap in recommended articles for separate initial articles

Word-Frequency-Based Model

- Problems with Word-Frequency-Based Models:
  - Different article length impacts reasonable similarity search
  - Exacerbated in combination with smoothing
  - For an article *a*, there are many articles more similar to *a* than *a* itself
- Other approach of finding surprising articles: entity correlation
  - On average, 8 entities per article
  - But: Correlations have strong power law distribution
  - Very few distinguishable entity correlations

Topic-Based model

- Second model (pursued via Master‘s thesis)
  - Calculate latent topics over all articles
  - For each document, the topic distribution can be calculated
  - Calculate topic correlations
  - Serendipitous documents will have
    - Unusual topic combinations (supported through initial user study)
    - Some of the initial article‘s topics
  - Evaluation in progress

Agenda

- Motivation
- Challenges

- Modeling Serendipity
  - Corpus
  - Word-Frequency-based
  - Topic-based

- Shift in research direction

New Direction

- Since topic-based model works best, shift focus a bit

- Use experience in the following fields:
  - Twitter analysis on virality (Master's thesis)
  - Handling of large text corpora, newspaper articles
  - Recommendation techniques
- Design with an application in mind

- Idea:
  - Combine Twitter virality prediction and news article suggestions

New Direction

- Given a Twitter user's tweets, an interest profile can be built and matched to newspaper articles

- Idea: Analyze user's followers as well and predict retweets
  - Step one: Predict number of retweets a user's link will generate
  - Step two: Recommend links that a user might want to share with his/her followers

Retweet prediction

- Step one: Predict number of retweets a user's link will generate
  - Crawl Twitter users' and their followers' tweets
    - Extract features (topics, followers, tweet behavior)
  - Traverse and crawl links, identify articles
    - Extract entities, key words, etc. from articles
  - Build interest profiles for users, compare to articles
  - Use machine learning to predict number of retweets for a link

  - Evaluation: Easy, compare predicted retweets with actual ones

Link recommendation

- Step two: Recommend links that a user might want to share with his/her followers
  - Obtain corpus of news articles
  - For each user, predict number of retweets for articles and recommend accordingly

  - Evaluation much harder
  - Solution: Build an application that people can use
    - Website that recommends articles that might be interesting to users and their followers
    - Directly contacting a user on Twitter with suggestions
  - See if the user will post link, measure retweets

Summary

- Recommending serendipitous newspaper articles
  - Word-Frequency-based approach: Unsatisfying
  - Topic-based approach: Promising
- New direction: Recommending links based for Twitter users
  - Learn which links are interesting and get retweeted
  - Predict retweets for unknown links