NTNU
Norwegian University of
Science and Technology

Online event recommendation

Retrieving and and analyzing Norwegian
screening events

Mads Løkken Paulsen

Master of Science in Computer Science
Submission date: June 2016
Supervisor: Jon Atle Gulla, IDI

Norwegian University of Science and Technology
Department of Computer and Information Science

Summary

As the available content on the Internet grows, it is becoming harder to find the relevant data you are looking for. A segment of the available content is future events (e.g. concerts, sports matches, public parties, book releases, cinema screening events, etc.) If a user wants to know what is happening next weekend in the nearest city, or is planning a vacation, he/she usually want to get a ranked list of personalized recommendations rather than scrolling through dozens of pages finding something interesting.

There is little relevant academic research done in this field in particular. The Thesis begins by briefly describing the *state-of-the art*, then presenting some already existing future event recommender systems and sources that provide Norwegian future events.

The system described in the thesis aims to extract Norwegian *screening events* and its relevant features. The data is modeled using JSON-LD with the Schema.org vocabulary. Then use this data to perform personalized recommendations. Two surveys were conducted to gather user preferences and find importance of relevant features. A simple recommender system were created, testing the performance of different features for a content-based filtering recommender. A collaborative filtering and a weighted hybrid recommender were created as well. And finally comparing the three approaches.

Sammendrag

Ettersom tilgjengelig innhold på Internett vokser, blir det stadig vanskeligere å finne relevante data du leter etter. Et segment av det tilgjengelige innholdet er fremtidige hendelser (f.eks konserter, idrettsarrangementer, offentlige fester, bokutgivelser, kino, osv.) Hvis en bruker ønsker å vite hva som skjer neste helg i nærmeste by, eller planlegger en ferie, han/hun ønsker som regel å få en rangert liste over personlige anbefalinger i stedet for å bla gjennom dusinvis av sider å finne noe interessant.

Det er lite relevant akademisk forskning gjort på dette feltet. Oppgaven begynner med en kort oversikt over *state-of-the art*, og deretter presentere noen allerede eksisterende anbefalingssystemer for fremtiden hendelses og kilder på internett som inneholder norske fremtidige hendelser.

Systemet er beskrevet i denne masteroppgaven tar sikte på å hente norske kinovisninger og dems relevante data. Dataene er modellert ved hjelp av JSON-LD med Schema.org vokabular. Deretter bruke disse dataene til å utføre personlige anbefalinger. To undersøkelser ble gjennomført for å samle brukerpreferanser og finne betydningen av relevant data. Et enkelt anbefalinssystem ble opprettet, dette for å teste et innholdsbasert anbefalingssystem ved å bruke forskjellige datafelter. En samarbeids filtrering og et vektet hybrid anbefalinssystem ble også opprettet. Til slutt ble de tre tilnærmingene sammenlignet.

Preface

This report presents the research and contributions for the work done in the master's thesis at Department of Computer and Information Science (IDI) Norwegian University of Science and Technology (NTNU). This report is the final delivery for the degree in Master of Science (Sivilingeniør) at NTNU.

The research and contribution performed, is a part of the larger SmartMedia project at IDI. The time frame of the work done is from January to June 2016. I would like to thank Professor Jon Atle Gulla for supervising this project, and providing valuable feedback.

Trondheim, June 20, 2016

Mads Løkken Paulsen

Table of Contents

List of Tables

List of Figures

Listings

Acronyms

ALS Alternating Least Squares. 15, 47

API Application Programming Interface. 22–24, 26, 31, 32, 61

CB Content-based filtering. 12–14, 19, 20, 42, 47, 55, 62, 63

CF Collaborative filtering. 12–14, 19, 20, 47, 58, 62, 63

EBSN Event-based social network. 9, 19, 21, 24

IDI Department of Computer and Information Science. iii, 4

IRI Internationalized Resource Identifier. 11

JSON JavaScript Object Notation. 11

MSE Mean Squared Error. xi, 16

nDCG Normalized Discounted Cumulative Gain. xi, 17

NTNU Norwegian University of Science and Technology. iii, 3, 4

RMSE Root Mean Squared Error. 16

SGD Stochastic Gradient Gescent. 14, 47

TF-IDF Term Frequency-Inverse Document Frequency. xi, 12, 15, 65

TOS Terms of Service. 23

URL Uniform Resource Locator. 32, 36, 37

XML Extensible Markup Language. 12

XPath XML Path Language. 12, 31

Part I

Introduction

Chapter 1

Introduction

This chapter is an introduction to the project as a whole, and will give an outline of the background and motivation, the research context, state the goals and questions, state the contributions described in this thesis, and lastly give a structure of the report.

1.1 Background and Motivation

The need for future event discovery is growing as the available events on the web are getting vaster. Events are usually presented on a wide variety of sources in different formats, and it is not easy to find relevant events if a person want to find an interesting event near a particular location and date-time. Someone might want to plan a trip to another place or discover what is happening the next weekend. There are already some solutions to this problem available, but with limited exploratory options and contains small amounts Norwegian event data. There is also little academic research performed in this field.

The motivation for this thesis is to gather relevant future event data for a particular sub-domain of future events. For the selected sub-domain there will be performed measures of basic recommendation approaches to set a baseline of recommendation techniques, and future research into the topic of future event recommendations.

1.2 Research Context

This report ia a mandatory delivery for the course TDT4900(Computer Science, Master's Thesis)[1]. This is mandatory for all student in Master of Science in Computer and Information Science at NTNU. The report is a part and further extension of the NTNU SmartMedia Program(subsection 1.2.1).

[1] www.ntnu.edu/studies/courses/TDT4900/2016

1.2.1 NTNU SmartMedia Program

The NTNU SmartMedia Program[2] is led and run by Professor Jon Atle Gulla[3] at IDI, NTNU. The program are currently developing a mobile news recommendation system, with further interest in expanding into the domain of feature event recommendation. The project is investigating into exploratory recommendation, where the reader can compose its own search strategies with help from the recommender system. With this type of recommendation, the idea is that a user can give better feedback consistent with the users' actual interests.

1.3 Goals and Research Questions

The goal for this project is to deliver a proposed data structure of future events, a way to retrieve them and integrate it with the SmartMedia Program. As there is many different event types, it was decided that screening events was a good starting point.

And thus the following questions arise:

RQ1 *How can we in real-time identify the location, date and time of all relevant movies on Norwegian cinemas?*

RQ2 *Which features can we automatically extract for these movies, and to what extent are these features relevant for movie recommendations?*

RQ3 *How do collaborative filtering, content-based recommendations and hybrid recommendation strategies compare with the features retrieved?*

1.4 Research Contributions

This thesis describes three main contribution that will aim to answer the three research questions.

And is as follows:

C1 Creating a web scraper for collection of screening events, corresponding to RQ1 and the first part of RQ2.

C2 Making a survey of cinema movie interest and evaluate what features people themselves deemed important (RQ2). There is also a follow up to gather test data for RQ3.

C3 Evaluate different recommendation techniques in accordance with RQ3. And to evaluate relevance of obtained features as in RQ2.

[2] research.idi.ntnu.no/SmartMedia/

[3] www.idi.ntnu.no/~jag/

1.5 Thesis Structure

The thesis structure is as follows:

Part 1 - Introduction :

Chapter 1 - Introduction This chapter is an introduction to the project as a whole, and will give an outline of the background and motivation, the research context, state the goals and questions, state the contributions described in this thesis, and lastly give a structure of the report.

Part 2 - Preliminary Study :

Chapter 2 - Theoretical Background This chapter will describe related theory used through this report.

Chapter 3 - Related Work This chapter will present related work to future event recommendation and movie recommendation, both in academia and the industry.

Chapter 4 - Event Sources This chapter will present relevant sources for future event recommendation.

Chapter 5 - Tools Used This chapter will present programming tools used in part 3.

Part 3 - Contribution :

Chapter 6 - Data Retrieval This chapter will describe the process of the we scraper for data retrieval.

Chapter 7 - Survey This chapter will describe how the survey is performed.

Chapter 8 - Screening Event Recommendation This chapter will describe how the different recommendation techniques are performed on the data gathered from the previous chapters.

Part 4 - Evaluation :

Chapter 9 - Results This chapter will present the results derived from the three contribution chapters in part 3.

Chapter 10 - Discussions This chapter will discuss the result presented in the previous chapter.

Chapter 11 - Conclusions This chapter will conclude the findings per research question.

Chapter 12 - Future work This chapter will describe some proposed future work related to future event recommendation.

Part II

Preliminary study

Chapter 2

Theoretical Background

This chapter will describe the theoretical background of technologies used in this project.

2.1 Future Events

Events are characterized by having specific locations and times assigned, ass well as a description of the event. Future events are the type of events that has the temporal data set to a future date and time. As to be discovered in chapter 4, these three fields are usually what is used for an event item at various event providers. But events could also have other kinds of explicit meta data as shown in chapter 6. There are different event types, like sport-events (i.e. football matches), music-events (i.e. concerts), education-events(i.e. scientific talks), etc. The different event types might not have all the same properties, as a football match would have the teams listed, but a concert does not have any teams.

Events are *one-and-only* items, but the content could in some cases be the same(reoccurring events). Movies at cinemas and different concerts on a concert tour would both have the same creative works performed, but with different time slots and/or locations.

2.1.1 Event Based Social Networks

Event-based social network (EBSN) was first defined in Liu et al. (2012). Users in EBSNs has two types of social interactions: online and offline. The first is when users use interactions, such as sharing thoughts about the events in social event groups or pushing calendars to their followers. The second interaction is when users physically meet at the place and time, this one is represented by attended events amongst users. Figure 2.1 illustrates two slightly different types of EBSNs. chapter 4 will look at various event sources, some of which are EBSNs.

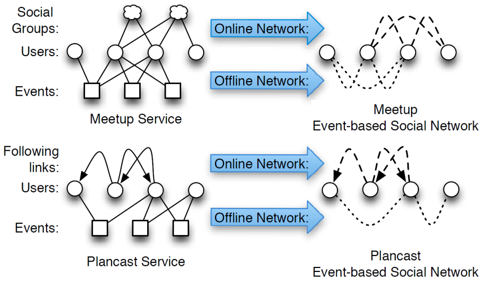


Figure 2.1: Network examples from Liu et al. (2012)

2.2 Linked Data

Linked data [1] is a way of structuring data, to create a network of machine-readable data across web sites, based on standards. It makes it possible for an application to start at a piece of Linked Data, follow the embedded links to use data hosted on other sites.

2.2.1 Schema.org

Schema.org is a collaborative community sponsored by Google[2], Microsoft[3], Yahoo[4] and Yandex[5]. Schema.org promotes structured data with a shared vocabulary for schemas, and thereby ontologies. It is used by over 10 million sites to markup web pages and e-mails.

IETF BCP 47 Standard

The IETF BCP 47 Standard [6] is used by the Schema.org vocabulary to define tagging for identifying languages. A tag could be as simple as ”fr” to identify French, but can also specify script type and narrow the set to specific locations and more. Ie: ”ar-Cyrl-CO” (Arabic, Cyrillic script, as used in Colombia), ”en-US”(Arabic, as used in United States) or ”nb-NO”(Norwegian Bokmål, as used in Norway).

ISO8601 Standard

The ISO8601 Standard [7] used by the Schema.org vocabulary to represent date, times, durations, etc. using numbers. The standard is an internationally accepted way of representation, and is created to tackle uncertainty.

The following ways of representations is used in this project:

YYYY-MM-DD Year, month and day. Example: ”2016-04-05”, to represent the 5th of April 2016.

[1] www.w3.org/standards/semanticweb/data
[2] www.google.com
[3] www.microsoft.com
[4] www.yahoo.com
[5] www.yandex.com
[6] tools.ietf.org/html/bcp47
[7] www.iso.org/iso/home/standards/iso8601.htm

YYYY-MM-DDTHH:MM Specify a particular time and date with year, month, day, hour and minutes. When time zone is not specified, the local time is assumed. Example: ”2016-05-31T18:45”, to represent the 5th of May 2016, at time 18:45 local time.

PT¡hours¿H¡minutes¿M A period using time with only hours and minutes. Example: ”PT1H45M”, representing a duration of 1 hour and 45 minutes.

2.3 JSON

JavaScript Object Notation (JSON) [8] is a lightweight data interchange format. It is easy for both humans and machines to read and write (parse and generate). The structure is represented using objects containing unordered name-value pairs. The object begins with left brace ”{” and is ended using right brace ”}”, each name-value pair uses colon ”:” between the name and value, and the pairs are separated using comma ”,”.

The values could be a string, number, object, array, true, false or null. Arrays contains zero or more value elements, the array starts with lefts square brace ”[” and ends with right square brace ”]”. Other value types will not be elaborated.

2.3.1 JSON-LD

JSON-LD [9] is a serialization based on JSON for Linked Data. A number of syntax tokens and keywords are specified, which of the following are used for the modeling of data in chapter 6:

@context The context defines the vocabulary of the object and sub-objects. It states the boundaries allowed for values in the different name-value pairs.

@id Identifier used for other objects to link to the identified object.

@type States the type, where the context specifies available fields and further linking.

: JSON keys and values separator for use of compact Internationalized Resource Identifiers (IRIs).

2.4 Web Scraping

Web scraping is a software technique used to extract information from web sites. There exists a number of techniques such as ”human copy-and-paste”, ”pre-built tools”, ”semantic annotation recognizing”, etc. In this project a ”HTML parser” is created (see: chapter 6). HTML parsing is often conducted when the web site is dynamically generated from underlaying structures like databases.

[8] `www.json.org`

[9] `www.w3.org/TR/json-ld`

2.4.1 XPath

XML Path Language (XPath) [10] is a language for addressing parts of an Extensible Markup Language (XML) document. It is used to extract nodes or a set of nodes from the selected document as used in chapter 6.

Example: ”//div[@id=’trailere ’]/div/a”, extracts all nodes named ”a” which is directly descended from ”div” descending from a node named ”div” having the ”@id” field set to ”trailere ”.

2.5 Recommendation Systems

The main goal of recommendation systems, is for recommending items to the user based on their preferences. An example is Amazon[11]: They have a lot of different books, but a user don’t have the whole day scrolling through the whole selection. A recommender system will use different methods to model the user, and use this to give a list of books that might suite the specific user based on previous purchases, other user reviews of the item and other types of feedback.

As to be described in subsection 2.5.2, future events behave a bit different then classical items. Such as only being relevant for a limited amount of time and explicit feedback often limited.

2.5.1 Content-Based

Recommendation using Content-based filtering (CB) is a user to item approach, looking at the user preferences and items similar to their preferences. CB approaches often used a vector of features of discrete values to learn and predict preferences. Feature extractions techniques such as TF-IDF (see: section 2.6). A CB approach needs to create and train individual models for each user.

2.5.2 Collaborative Filtering

Collaborative filtering (CF) have the upside of not needing to know much about the item, but rather what users think of the item. The only data needed is a user-item-rating triple, and sometime an optional time-stamp for when it was rated Based on reviews and other types of feedback from other users, a CF approach can recommend an item to a user. If two users are deemed similar, and user1 has given a review about item1, and user2 has not consumed that item yet, then the item might get recommended to that user. A CF recommender only creates and trains one model at a time for all users and items.

2.5.3 Hybrid recommender

A hybrid recommender uses the best of both worlds, combining multiple approaches into one final recommendation. There are some different techniques of how to combine the

[10] www.w3.org/TR/xpath

[11] www.amazon.com

multiple recommender approaches as shown in Table 2.1.

Hybridization method	Description
Weighted	The scores (or votes) of several recommendation techniques are combined together to produce a single recommendation.
Switching	The system switches between recommendation techniques depending on the current situation.
Mixed	Recommendations from several different recommenders are presented at the same time.
Feature combination	Features from different recommendation data sources are thrown together into a single recommendation algorithm.
Cascade	One recommender refines the recommendations given by another.
Feature augmentation	Output from one technique is used as an input feature to another.
Meta-level	The model learned by one recommender is used as input to another.

Table 2.1: Hybridization Methods from Burke (2002)

2.5.4 Cold-Start Problem

The cold-start problem is when something new is introduced to the recommendation system, namely users and/or items.

Sparsity in future event attendance feedbacks and reviews, they are prune to the cold-start problem. In the Amazon example [section 2.5], the recommendation could be based on previous reviews. But as for events, there are often little or none reviews about the item when it need to be recommended, because a user can't review an item before it is consumed. According to de Macedo and Marinho (2014), the lifetime range of events are usually between 5-100 days, with most of the attendance response in the last 20% of the lifetime.

Both CB and CF based approaches needs item ratings for a sufficient amount of items before the model can predict reliably. When a new user is introduced to the system, the model can not reason about what that particular user might like. Common approaches is to either make the users rate a set of diverse items until the recommender can make somewhat relevant personal recommendations. Another approach used, inter alia, by Microsoft's Matchbox recommender[Stern et al. (2009)] is to calculate user similarity based on the user profile (age, location, interests, gender, etc.) and recommend based on the similar user's ratings, then gradually switch to pure CF and/or CB.

CF is also prune to cold-start items, since it is recommending items based upon similar user's feedback of the particular item. Suggested approaches are to use a hybrid model with higher CB weighting for cold items until a sufficient amount of feedback is given, then gradually switch to potentially more optimal settings. CB approaches can also suffer to some decree from cold-start items, if new feature values are introduced, i.e. a particular movie genre that have not been present in any of the trained data.

2.5.5 Temporal Filtering

By selecting a day or time slot, the recommender can filter out events falling outside these time limits. As the SmartMedia Program uses an exploratory approach to the news domain, which a future event recommender will be a part of, temporal filtering would be one of the main techniques used for event recommendation. Temporal information is often used as a part of CB, but with the idea of users themselves selecting locations, there would be little gain using this information in CB and would rather be a binary selection before classification.

2.5.6 Location Based Filtering

Location filtering would have the same idea as for temporal properties, but with locations instead. Using either radius or predefined areas. A user might want to plan which events it wants to attend when traveling to other locations. Using this binary approach together with temporal filtering, the recommender need only to work with smaller subsets of the vast event domain.

2.5.7 Linear Regression

A simple CB technique is to use Linear Regression, which is an approach to model the relationship between the item features an the user feedback. The goal is to draw a line in the feature space, minimizing the distance error for the trained data points.

Stochastic Gradient Descent

Linear Regression can use Stochastic Gradient Gescent (SGD), which is a technique for minimizing the error of the regression line. Starting with random weights of the features creating the line, the derivative of the line would indicate which direction the weight values should be changed, after enough iterations, the derivative would reach zero and hence a local or global minimum.

2.5.8 Matrix Factorization

A common approach to model a CF recommender is to use Matrix Factorization. The user-item-rating model is factorized into two separate matrices with a N latent factors, as seen in Figure 2.2. The model will then reduce memory space for large matrices, from $U * I$ to $I * N + U * N$. A matrix with 1000 users an 10’000 items would be scaled down from 10’000’000 to 22’000 with two latent factors. The two resulting matrices can bee seen as one matrix with rows of items and hidden unknown features, and one with user columns with weights. To calculate the estimated rating for an item for a user, you multiply the item row with the user column.

$$A = XY$$

$$\begin{pmatrix} a_{1,1} & a_{1,2} & a_{1,3} & a_{1,4} \\ a_{2,1} & a_{2,2} & a_{2,3} & a_{2,4} \\ a_{3,1} & a_{3,1} & a_{3,3} & a_{3,4} \end{pmatrix} = \begin{pmatrix} x_{1,1} & x_{1,2} \\ x_{2,1} & x_{2,2} \\ x_{3,1} & x_{3,2} \end{pmatrix} \begin{pmatrix} y_{1,1} & y_{1,2} & y_{1,3} & y_{1,4} \\ y_{2,1} & y_{2,2} & y_{2,3} & y_{2,4} \end{pmatrix}$$

Figure 2.2: Matrix Factorization with two latent factors

Alternating Least Squares

To factorize a sparse matrix of many unknown values, Alternating Least Squares (ALS) [Koren et al. (2009)] can be used to learn and model the latent factors. The method aims to generate two matrices when multiplied will reduce the errors for the known values i the original matrix.

The method starts with initializing the first X row with the average rating values, and the rest with random numbers. Then calculate Y based on A and X, then change values in Y based on minimizing the least square error function when multiplying X and Y. When Y is minimized, X is recalculated based on A and Y. Then minimize and alternate until both X and Y satisfies the least square error function at the same time.

2.6 Feature Extraction

Feature Extraction is techniques used to extract discrete values from non-discrete data.

2.6.1 TF-IDF

TF-IDF is a feature extraction method to measure the number of times a word appears in a document and discount values for words that appear more frequently in all different documents. Figure 2.3 shows the calculation of TF-IDF weights. Words like ”the” and ”it” is usually present in almost all documents, and is then weighted low since the usually don’t provide much recommendation interest.

$$\begin{aligned} w_{i,j} &= tf_{i,j} * log(\tfrac{N}{df_i}) \\ w_{i,j} &= \text{weight of i in j} \\ tf_{i,j} &= \text{number of occurrences of i in j} \\ df_i &= \text{number of documents containing i} \\ N &= \text{total number of documents} \end{aligned}$$

Figure 2.3: TF-IDF function

2.7 Feedback

When a user interact with an applications it could leave various sorts of different feedback that could give valuable information about how interesting items are, how they use the application, etc. An example of feedback is shown i Table 2.2.

Feedback activity		Feedback value
Click on	’I like this’	1.0
Share on	Facebook/Twitter	0.9
Click on	Itinerary	0.6
Click on	Print	0.6
Click on	’Go by bus/train’	0.6
Click on	’Show more details’	0.5
Click on	’Show more dates’	0.5
Mail to	a friend	0.4
Browse to	an event	0.3

Table 2.2: Feedback used in Dooms et al. (2011)

2.7.1 Direct Feedback

Explicit feedback is usually known to the user. It could be when the user share an event, boy tickets or save the event to their calendar.

2.7.2 Indirect Feedback

Indirect feedback is often not known to the user. This could be when users decides to show more detailed information, view a related trailer or browse directly to the source.

2.8 Metrics

Different metrics are used to evaluate how good a recommender system performs. This section will present the performance metrics used for evaluation in this report.

2.8.1 MSE

MSE calculates the deviation of the predicted value compared to the actual value (Figure 2.4). By squaring the error prediction very far from the actual value is punished harder then many close by. For an unbiased estimator it MSE is known as the variance. MSE is not applicable to indirect feedback, since we do not have actual preference data, but rather estimated preferences.

$$MSE = \frac{1}{n} \sum_{i=1}^{n} (\hat{Y_i} - Y_i)^2$$

Figure 2.4: MSE function

RMSE

Root Mean Squared Error (RMSE) is the root of the MSE, and places more emphasis on larger deviations. For an unbiased estimator the RMSE is known as the standard divination.

2.8.2 nDCG

nDCG is a ranking metric that can be used to calculate how good multi-class ranking is. Where other metrics is calculated based on the document being relevant or not, nDCG can also calculate documents that are ranked with more classes like ”relevant”, ”somewhat relevant”, ”not relevant” and more.

The discounted cumulative gain is first calculated using the predicted rating with the function given in Figure 2.5, which is a version of the equation with a stronger emphasis on retrieving relevant documents.

To calculate nDCG the discounted cumulative gain is divided by a calculated optimal discounted cumulative gain (see: Figure 2.6).

p is the size of the ranked list and nDCG is often stated as nDCG@p, i.e. nDCG@6 for evaluating the top six ranked documents compared to an optimal top six.

$$DCG_p = \sum_{i=i}^{p} \frac{2^{rel_i} - 1}{log_2(i+1)}$$

Figure 2.5: DCG function

$$nDCG_p = \frac{DCG_p}{IDCG_p}$$

Figure 2.6: nDCG function

Chapter 3

Related Work

This chapter will try to summarize the relevant academic work performed with focus on future event recommendation, with the last section briefly describe some existing event recommendation services. Even if there is allot of relevant work connected to movie recommendation, the focus of this paper is still future events, and there has not been given much consideration to movie recommendation in particular when conducting the preliminary study

3.1 Hybrid Approach

A hybrid approach to future event recommendation was first[1] described in Cornelis et al. (2005), and the ideas later used in related works. The model is using both CF and CB, modeling item and user similarities as fuzzy relations. Described as: ”Recommending future items if they are similar to past ones that similar users have liked.”(Cite fromCornelis et al. (2005)).

3.2 Recommending in Event-Based Social Networks

EBSNs as first defined in Liu et al. (2012) is used i most newer researches Zhang et al. (2013); de Macedo and Marinho (2014); Qiao et al. (2014); Ji et al. (2015); Purushotham and Kuo (2015); Macedo et al. (2015). The recommending approaches uses social connections, co-attendance on the same event, membership of the same social group, friendships, etc. This approach together with item similarity, temporal and geographical awareness is evaluated by Macedo et al. (2015) with good results.

[1] To my knowledge.

3.3 Pairwise Approach

A CB approach given by Minkov et al. (2010) is modeling the user by giving it a selection of two items, where the most preferred event is selected and a learner will reason an tune parameters.

3.4 Using Linked Data

Using semantics to better describe events is used in Khrouf and Troncy (2013) and Zhang et al. (2013). By modeling event data this way the recommender could more easily explore and reason over data, as it is possible to link with other more descriptive datasets i.e. linking to a well established database like DBPedia[2] to compare artist. And as stated by Zhang et al. (2013) it is impractical to store all related data in one ontology. This is also partly looked at by Kayaalp et al. (2009), stating that a person attending an event by a band with one particular genre might also like another band with the same linked genre.

3.5 Ranking From Implicit Feedback

Ranking a set of items form implicit feedback is described in Rendle et al. (2009) and their work is the baseline that Macedo et al. (2015) compare against. Recommending by the much easier to collect implicit feedback rather then explicit given by the user, and the work provide generic algorithms and optimization creation.

3.6 User-centric Evaluation

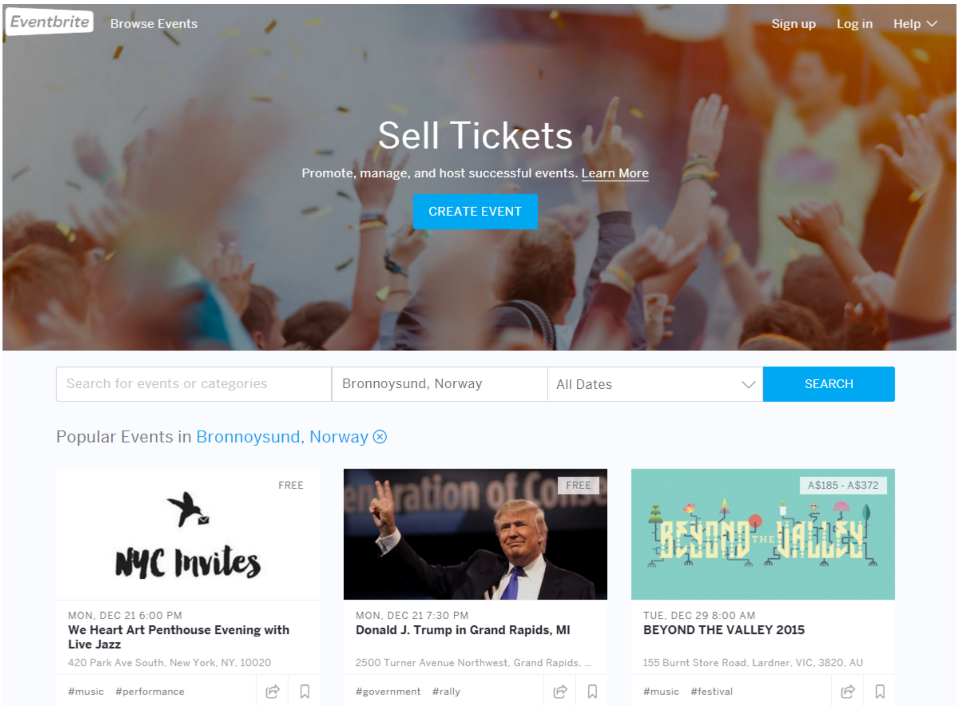
Paper Dooms et al. (2011) compares different state-of-the art[3] recommender algorithms for event recommendation. Giving users the ability to test randomly selected algorithm, it shows that a hybrid approach performs best in user satisfaction. The result show that the user satisfaction of a recommender is predicted by how well recommendations match the users' interests. User based CF performed better then CB, and the hybrid approach in this paper was just a selection of the top three from each filter.

3.7 Existing Future Event Recommendation Services

There are already some solutions available for event recommendation. This section will describe them in short and how they work. Some of these could also be used as sources, and would be elaborated as that in chapter 4.

[2] `www.dbpedia.org`

[3] State-of-the art in 2011.



Sell Tickets

Promote, manage, and host successful events. Learn More

Search for events or categories | Bronnoysund, Norway | All Dates

Popular Events in Bronnoysund, Norway

MON, DEC 21 6:00 PM
We Heart Art Penthouse Evening with Live Jazz
420 Park Ave South, New York, NY, 10020
#music #performance

MON, DEC 21 7:30 PM
Donald J. Trump in Grand Rapids, MI
2500 Turner Avenue Northwest, Grand Rapids, ...
#government #rally

TUE, DEC 29 8:00 AM
BEYOND THE VALLEY 2015
155 Burnt Store Road, Lardner, VIC, 3820, AU
#music #festival

Figure 3.1: Eventbrite

3.7.1 Facebook Events For You

Facebook events[4] are a great source for events, and ”Events For You” was introduced in 2014. They look at the users’ information to give a recommendation i.e. liked pages, groups, communities, events attended by friends etc. Much like in EBSN approaches. The user can’t input other locations or preferences, just nearby. This recommender is being tested in parts of the world and under development.

3.7.2 Eventbrite

Eventbrite[5](Figure 3.1) recommends events using location, topic and type filters set by the user. It also tracks past attended events and information from their social graph. They use themselves as a source and ticket provider (see section 4.1).

[4] `events.fb.com`

[5] `www.eventbrite.com`

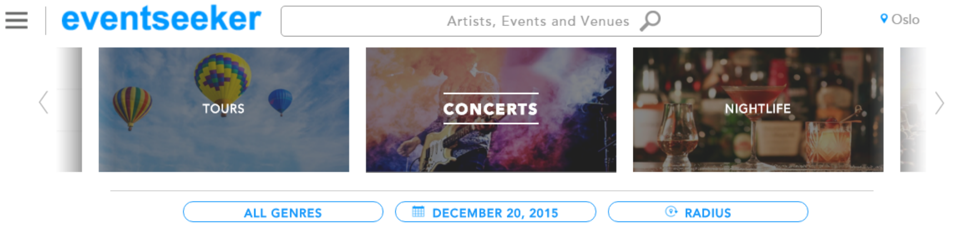


Figure 3.2: Eventseeker

3.7.3 Eventseeker

Eventseeker[6](Figure 3.2) can be used through their web-page or the application. They use the Wcities[7] private Application Programming Interface (API), and uses the users Facebook data(using similar approaches to what described in subsection 3.7.1), scans the cellphone for music and more to give personalized events. They get events from Facebook and their 150+ partners such as Ticketmaster, StubHub, Ticketfly, Eventim, Seatwave, Fnac etc.

The user can share, buy tickets, follow performers, discover based on location.

3.7.4 Other

Nearify Application that recommend based on location and user explicitly following event types.

Eventful Not showing me any events, but use location radius filter, movies as a separate recommendation, and the ability to get events from a variety of event types.

Eventster Recommender application for iPhone.

Upcoming Bought by yahoo as `events.yahoo.com` later shut down. Now sold back to the original creator and under development after a fund rising campaign, showing interest in event recommendation.

[6] `www.eventseeker.com`

[7] `www.wcities.com`

Chapter 4

Event Sources

This chapter will investigate some event sources available, to what extent do they use relevant meta data and how accessible they are. It is also worth mentioning that because of legal issues, some providers only allow personal use of content available on their web page and prohibit usage of data mining tools if not agreed upon e.g. Billetservice.no Terms of Service (TOS)[1].

From related work there are possible to extract some event sources used. Since the SmartMedia project is located in Norway there are also a greater emphasis on the Norwegian sources, this chapter will therefore also look at some of the bigger Norwegian event sources, and some smaller local providers.

4.1 Eventbrite

Eventbrite[2][Figure 4.1] is the largest self serviced ticket platform in the world. Organizers has the ability to create their own events, with a small fee paid to Eventbrite for the promotion service. This source does not contain allot of Norwegian events.

Figure 4.1: Eventbrite logo

The meta data is represented mostly using Schema.org vocabulary [subsection 2.2.1], using most properties. The event typing is not always consistent i.e. sometimes concerts are typed as visual arts or plain event.

Provides an open API.

[1] www.billettservice.no/help/Footer/billettservice.termOfUse.EN.htm
[2] www.eventbrite.com

4.2 Facebook

Facebook[3][Figure 4.2], is the larges social network in the world. People and organizations has the opportunity to create events for free. It contains allot of small local events, but also larger ones to some extent.

The events are represented using Facebooks' own Open Graph vocabulary [4]. Giving description, name, place, start time and possible ticket link.

Provides an API that requires an user access token to search for events.

Figure 4.2: Facebook logo

4.3 Meetup

Meetup[5][Figure 4.3] is an EBSN often much used in relevant academic work on event recommendation. Users can join groups that are creating events, they can share and comment on the events. Event can be closed to outside users from the group. Most of the groups based in Norway seems to be software related.

Meetup uses the deprecated data-vocabulaty.org[6]. Using the API there is possible to get description, duration, host of event, ticket info, group hosting the event, photos, max limit of attendance, attending users, status and more.

Provides an open API.

Figure 4.3: Meetup logo

4.4 Billettservice

Billettservice[7][Figure 4.4] is Norway's leading event ticket provider, run by Ticketmaster[8]. Organizations using Billettservice often promote the event creating Facebook events with links to the Billetservice ticket site.

Events are categorized into types with sub-types. Using date, time, venue, name, ticket information, description and image. The categories is not stated as meta data in the source, but somewhere internally it is.

Figure 4.4: Billettservice logo

[3] www.facebook.com

[4] http://ogp.me/

[5] www.meetup.com

[6] http://www.data-vocabulary.org/

[7] www.billettservice.no

[8] http://www.ticketmaster.com/

4.5 Filmweb

Filmweb[9][Figure 4.5] is a Norwegian information provider of almost all cinematic movie screenings in Norway. It lists almost all cinemas in Norway and makes it possible for users to find films and to buy tickets.

Figure 4.5: Filmweb logo

Movie objects are described with description, genre, length, age restrictions, premiere date in Norway, profiled actors, and more. Users can leave feedback to rate the movie, and Filmweb provides linking to reviews from the film critics in media. The event itself is providing genre, short description, date-time, with location and linking to tickets.

Filmweb provides user with filtering based on location and/or cinema and dates, as well as a selected set of genres. On the front page they some different global recommendations, such as time relevant movies, premieres, upcoming and popular movies based on user ratings.

4.6 Hoopla

Hoopla[10][Figure 4.6] is a smaller relatively new Norwegian ticket provider and promoter.

Hoopla is representing the items with name, description, image, location, ticket site, start and end date-time, and some more unrelated fields. Using Open Graph vocabulary.

Figure 4.6: Hoopla logo

4.7 Trdevents

Trdevents[11][Figure 4.7] is an event page containing only local events happening in the Norwegian city of Trondheim.

The events contains name, category, start date-time, venue details, ticket information, recurring events, image, target audience ages, language, description. Using Open Graph vocabulary for some parts.

Figure 4.7: Trdevents logo

[9] www.filmweb.no

[10] www.hoopla.no

[11] trdevents.no

4.8 Viagogo

Viagogo[12][Figure 4.8] is the worlds larges event ticket marketplace.

Figure 4.8: Viagogo logo

Viagogo is using Schema.org representation for the events. Using offers, name, start date-time, name of event, describing it, linking to venue. If an application want to know who is playing a football match, it looks like it need to reason about the name of the event as the opposing teams are not stated.

Provides an open API.

4.9 Other Sources Worth Mentioning

Last.fm Has an open API, listing concerts, much information about artist, also includes functions for similarities.

10times.com Not much used in Norway.

Wcities.com Used by their own recommender as described in subsection 3.7.3, and provides a closed API.

Opraen.no Using Schema.org vocabulary [subsection 2.2.1], using most properties but all event are of type event. performers are also represented by Schema.org.

Billettportalen.no A Norwegian ticket distributer.

IMDb.com While IMDb does not contain future event, it has an open API that provides much meta data about movies and can be used for screening events.

[12] www.viagogo.com

Chapter 5

Tools Used

This chapter will describe the tools used in the project. Tools used are Java compatible as all code is written in Java.

5.1 Selenium-java

Selenium[1][Figure 5.1] is a set of technologies used for automated browser control, written i Java. Selenium is mainly made for testing of web applications, but can be used for all automation processes of a browser.

Figure 5.1: Selenium logo

5.2 PhantomJS

PhantomJS[2][Figure 5.2] is a headless browser (does not draw to screen), used to run JavaScript generated content for the developed web scraper in chapter 6. A headless browser is preferred for automatic web processes, as it does not use resource time for displaying content on screen.

Figure 5.2: PhantomJS logo

[1] www.seleniumhq.org

[2] phantomjs.org

5.3 Apache Spark

Apache Spark[3][Figure 5.3] is a fast, in-memory date processing engine which includes different development frameworks, including a machine learning framework named ”Spark MLlib”.Spark MLlib includes a variety of machine learning techniques which is used in this project.

Figure 5.3: Apache Spark logo

5.4 Apache Maven

Apache Maven[4][Figure 5.4] is used to include tools used into the project, and keep them up to date. This is i done by including the dependencies in the ”pom.xml” file as shown in Listing 5.1.

Figure 5.4: Apache Maven logo

```
<dependency>
    <groupId>org.seleniumhq.selenium</groupId>
    <artifactId>selenium-java</artifactId>
    <version>2.52.0</version>
</dependency>
```

Listing 5.1: Maven dependencies example

[3] spark.apache.org

[4] maven.apache.org

Part III

Contributions

Chapter 6

Data Retrieval

This chapter will describe how a web scraper was created to get most planned future screening events in Norway. The first section will briefly state why this is chosen, followed by the collection of location data and finally the retrieval of screening events and movies.

6.1 Introduction

There is no singular location to get all future events in Norway. Filmweb[1] was selected, since it has information about most of planned screening events in Norway. Due to the lack of a public API, it was created a web scraper to gather Screening events. All data retrieval is done in Java with Maven (section 5.4) to keep other used Java tools up to date. Filmweb uses the meta field ”robots” with the content ”index, follow”, stating it is ok to index the whole website.

6.2 Cinema Theater Retrieval

All cinemas is retrieved from ”`www.filmweb.no/program/velgsted/`”, using XPath to retrieve all scripts. Then search for the script containing *”var loclist”*. This variable contains an array with all locations, where each location has a list of theater names as seen in Listing 6.1.

All locations are formated into JSON-LD objects (subsection 2.3.1), and using the ”Place” type from the Schema.org vocabulary (subsection 2.2.1), a final location object can be seen in Listing 6.5. All locations is containing one or more theater place objects which are the places featuring all the various different screenings obtained in section 6.3.

The geographical coordinates for all the locations are gathered using the Google Maps Geocoding API [2]. A request for all the obtained location names are sent using ”LOCA-

[1] `www.filmweb.no`

[2] `developers.google.com/maps/documentation/geocoding`

```
var locList =
[<location object 0>,...,{”location”:”Trondheim”,”theaters”:[”Nova”,”
    ↪ Prinsen”],”label”:”Trondheim”,”searchLabel”:”TRONDHEIM NOVA PRINSEN
    ↪ ”},...,<location object n>];
```

Listing 6.1: var loclist (minimized)

TION_NAME,+Norway” as seen in Listing 6.2. The resulting response is seen in Listing 6.3, and is a JSON object with a result array and a status string. The longitude and latitude is extracted and put into its appropriate ”GeoCoordinates” object in the ”geo” field of the location. In rare cases where the API returns a result with more locations, the first one is used (The best match is placing API as the first element).

```
https://maps.googleapis.com/maps/api/geocode/json?Trondheim,+Norway&key=<
    ↪ API_KEY>
```

Listing 6.2: Geocode request Trondheim

As seen in Listing 6.3, the returned location coordinates are usually an ”APPROXIMATE”, and not always a perfect representation of the actual theater locations. The next step to get more accurate positions for the theaters, was to manually search and find coordinates using Google Maps [3]. It was also considered obtaining the locations automatically, but most of the cinemas are not tagged as a theater in Google Maps, and it is a one time task, although cumbersome. After a search for the theater, a resulting Uniform Resource Locator (URL) will contain the correct coordinates for the theater. Data is in the form ”BASE_URL/NAME/@LATITUDE,LONGITUDE” as seen in Listing 6.4. Many of the theaters are serviced by Bygdekinoen [4], and the venues are not full time cinematic theaters. It could be a local school gymnasium, youth house, communal cultural building, community house, etc. Due to a lack of addresses stated on Bygdekoinoen some of the coordinates are based upon guesses, looking at the theater name. Sometimes in doubt it was also done some extensive searches for an address on the ”theater” Facebook page or communal homepage. A full geographical plot of all the cinema theaters is shown in Figure 6.1.

[3] www.google.no/maps

[4] www.bygdekinoen.no

```
{
   ”results” : [
      {
         ”address_components” : [
            {
               ”long_name” : ”Trondheim”,
               ”short_name” : ”Trondheim”,
               ”types” : [ ”locality”, ”political” ]
            },
            ...
         ],
         ”formatted_address” : ”Trondheim, Norway”,
         ”geometry” : {
            ...,
            ”location” : {
               ”lat” : 63.4305149,
               ”lng” : 10.3950528
            },
            ”location_type” : ”APPROXIMATE”,
            ...
         },
         ...
      }
   ],
   ”status” : ”OK”
}
```

Listing 6.3: Geocode response Trondheim (minimized)

```
https://www.google.no/maps/place/Nova+Kinosenter+(Trondheim+Kino)/@63
    ↪ .4331938,10.3997873,<MORE_DATA>
```

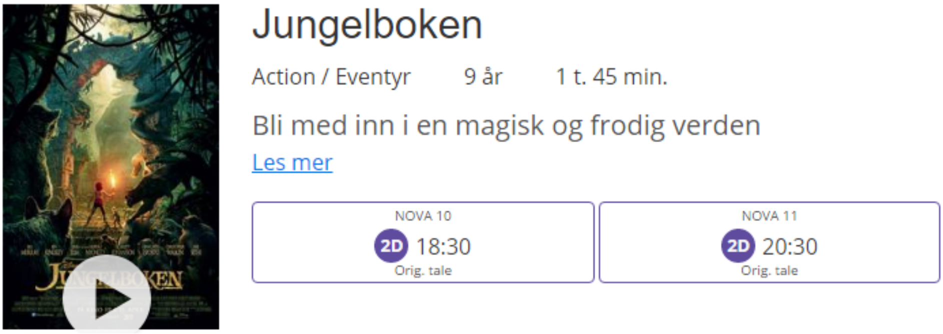
Listing 6.4: Google Maps search

6.3 Screening Event Retrieval

The screenings on Filmweb are rendered in JavaScript at ”`http://www.filmweb.no/program/`”. Selenium (section 5.1) with PhantomJSDriver (section 5.2) needs to be used. Without using an JavaScript engine, only the following text will be shown: *”Kinoprogrammet krever javascript for å kunne vises.”(English: Cinema program requires javascript to be displayed.).*

```
{
    ”@context”: ”http://schema.org”,
    ”@type”: ”Place”,
    ”@id”: ”Trondheim”,
    ”name”: ”Trondheim”,
    ”containsPlace”: [
        {
            ”@type”: ”Place”,
            ”@id”: ”Nova”,
            ”name”: ”Nova”,
            ”containedInPlace”: ”Trondheim” ,
            ”geo”: {
                ”@type”: ”GeoCoordinates”,
                ”latitude”: ”63.4331938”,
                ”longitude”: ”10.399782”
            }
        },
        ...,
        <THEATER PLACE N>
    ],
    ”geo”: {
        ”@type”: ”GeoCoordinates”,
        ”latitude”: ”63.4305149”,
        ”longitude”: ”10.3950528”
    },
}
```

Listing 6.5: Place JSON-LD (minimized)



Jungelboken

Action / Eventyr 9 år 1 t. 45 min.

Bli med inn i en magisk og frodig verden

Les mer

Figure 6.2: Rendered article

The web scraper start the process by getting the available dates for a given cinema theater sending the location name as a parameter in the request as seen in Listing 6.6. The resulting HTML page generated by JavaScript, will either contain a select class named *”dropdown dateOptions”* (Figure 6.3) with all dates having planned future events for the location. Or if the location has few planned screenings, all dates will be rendered in one request, and the XPath search will return zero elements. When *”dropdown dateOptions”* is not present, only todays date is returned, as all planned events will be shown anyway.

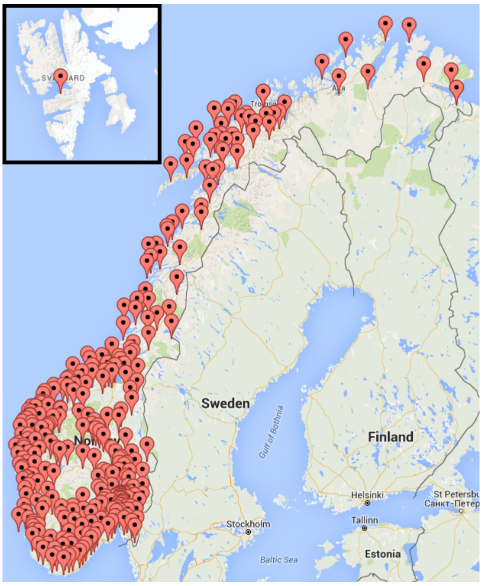
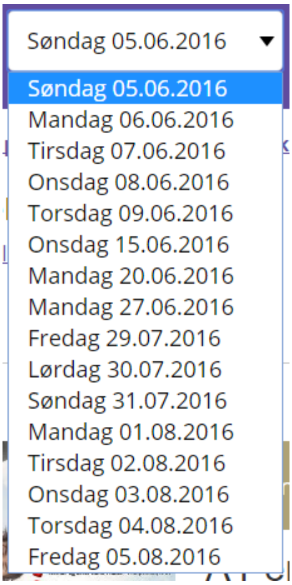


Figure 6.1: Cinema theater geographical plots

```
http://www.filmweb.no/program/#location=Trondheim
```

Listing 6.6: Screenings date request



Søndag 05.06.2016

Søndag 05.06.2016
Mandag 06.06.2016
Tirsdag 07.06.2016
Onsdag 08.06.2016
Torsdag 09.06.2016
Onsdag 15.06.2016
Mandag 20.06.2016
Mandag 27.06.2016
Fredag 29.07.2016
Lørdag 30.07.2016
Søndag 31.07.2016
Mandag 01.08.2016
Tirsdag 02.08.2016
Onsdag 03.08.2016
Torsdag 04.08.2016
Fredag 05.08.2016

Figure 6.3: dropdown dateOptions

After the search able dates are retrieved, the returned array with the dates are iterated getting future events using location, date and theater as shown in Listing 6.7. The response is parsed with XPath, to get al nodes with the id named *"programContainer"* using XPath string shown in Listing 6.8, only the responses with multiple dates shown on the same screen will have more then one program containers. If there is no container at all, the script has not loaded, and is put in a list to reload later. If the second returned node class name is *"noShows row"* there are no screening events for the request. A response could contain more dates, and not even the date that is requested, there might be no screening at that particular date. To get the right date, the first child node is parsed into ISO8601 standard (section 2.2.1) and is used a the date for all screening events retrieved from that particular program container.

```
http://www.filmweb.no/program/#location=Trondheim&date=31.05.2016&theater=
    ↪ Nova
```

Listing 6.7: Screenings request

```
//div[@id='programContainer']/div/div
```

Listing 6.8: XPath programContainer, event nodes by day

After obtaining the node date, it is removed and the remaining child nodes are articles, each article containing one or more screening event of the same movie at different screens, times, video format and/or language (Figure 6.2). The movie title node contains an attribute with the URL to a work presented, if the movie is not already parsed as described in subsection 6.3.1 it will be done. All the data present will be put into a *"ScreeningEvent"* object as shown in Listing 6.9.

Typical age range is used as the age restriction e.g. 9 years is "9-" and *"Tillatt for alle"(English: Allowed for all)* will be "0-". The subtitle language is set to "nb-NO" if "Orig. tale"(English: Original speech) is stated and the movie language is not Norwegian, and "utekstet"(English: not texted) is not specified. The "inLanguage" field should follow the IETF BCP 47 standard (section 2.2.1), but the translation is not done at this given moment. The "offer" field contains the ticket site URL for the particular event. When no URL is present in the purchase node, it is either because the screening event has already started, or the tickets have to be bought at the cinema theater entrance.

```
{
    ”@context”: ”http://schema.org”,
    ”@type”: ”ScreeningEvent”,
    ”name”: ”Jungelboken”,
    ”subtitleLanguage”: ”nb-NO”,
    ”videoFormat”: ”2D”,
    ”startDate”: ”2016-05-31T18:45”,
    ”duration”: ”PT1H45M”,
    ”workPresented”: ”http://www.filmweb.no/film/article1139116.ece”,
    ”typicalAgeRange”: ”9-”,
    ”inLanguage”: ”engelsk”,
    ”description”: ”Bli med inn i en magisk og frodig verden”,
    ”image”: ”http://www.filmweb.no/incoming/article1262844.ece/
    ↪ representations/h/Jungelboken%20(plakat)”,
    ”url”: ”http://www.filmweb.no/program/#location=Oslo&date=31.05.2016&
    ↪ page=pageShowsForDay&theater=Colosseum”,
    ”offer”: ”https://bestill.nfkino.no/BillettSystem/
    ↪ ChooseTicketCategories?firmId=3&showId=10034463”,
    ”location”: {
        ”@type”: ”Place”,
        ”name”: ”Colosseum 4”,
        ”containedInPlace”: {
            ”@type”: ”Place”,
            ”@id”: ”Colosseum”
            ”name”: ”Colosseum”,
            ”containedInPlace”: ”Oslo”,
            ”geo”: {
                ”@type”: ”GeoCoordinates”,
                ”latitude”: ”59.929626”,
                ”longitude”: ”10.7082422”
            }
        }
    }
}
```

Listing 6.9: ScreeningEvent JSON-LD

6.3.1 Movie Retrieval

Movies ar retrieved using the references given by the screening events, using the page source as movie id. All relevant data nodes are collected using Xpath queries as seen in Listing 6.10. Some of the data is stated at multiple locations, and sometimes not present at all. In particular the movie facts varies in data, and there was not found a suitable translated counterpart in the Schema.org[subsection 2.2.1] vocabulary for the following data: *”Distribusjon”, ”Begrunnelse”, ”Video distribusjon”, ”Egnethet” and ”Medvirkende”(”Actors” in documentaries).* There is also production year which is not put into the model as the Norwegian premiere date is used instead.

Persons are retrieved as person types [Listing 6.11] (with or without URL depending on what is present) and organizations as the organization type [Listing 6.11], they ar put into the model depending on the field they are stated in. All trailers are put into separate video objects [Listing 6.13], with the URL to the trailer on Filmweb.

```
//div[@class='filmomtale full']/h1
//div[@class='largeTitle']
//meta[@name='title']/@content
//meta[@name='description']/@content
//meta[@property='og:image']/@content
//div[@class='userAvg']/span
//div[@class='userAvg']
//div[@class='ingress']
//div[@class='bodytext']
//ul[@class='facts completeFacts']/li
//div[@id='trailere']/div/a
//div[@id='anmeldelser']/a
```

Listing 6.10: XPaths for movie data

```
{
    "@type": "Person",
    "name": "Jon Favreau",
    "url": "http://www.filmweb.no/profil/article858231.ece"
}
```

Listing 6.11: Person type object

Reviews are put into review objects [Listing 6.14], containing the publisher organization and rating type. Ratings are usually from 1 to 6, if worst rating is ommited it defaults to one as stated in the Schema.org specifications. Edge cases were discovered in later stages of the type ”7 / 10”, and are presumed to have the default lowest value of 1 as all other ratings.

The final movie objects is shown in Listing 6.15. Dates and durations is formatted into the ISO 8601 date format [section 2.2.1] and different textual descriptions put into a string list. The ”inLanguage” field should be formatted into the IETF BCP 47 standard [section 2.2.1], but was not implemented due to all different cases and uncertenties of the format used by Filmweb.

```
{
    "@type": "Organization",
    "name": "Walt Disney Pictures"
}
```

Listing 6.12: Organization type object example

```
{
    ”@type”: ”VideoObject”,
    ”name”: ”Trailer 2”,
    ”url”: ”http://www.filmweb.no/trailere/article1270584.ece?autoplay=
    ↪ true”
}
```

Listing 6.13: VideoObject type example

```
{
    ”@type”: ”Review”,
    ”publisher”: {
        ”@type”: ”Organization”,
        ”name”: ”NRK P3”
    },
    ”reviewRating”: {
        ”@type”: ”Rating”,
        ”bestRating”: ”6”,
        ”ratingValue”: ”5”
    },
    ”url”: ”http://p3.no/filmpolitiet/2016/04/jungelboken/”
}
```

Listing 6.14: Review type object example

```
{
    "@context": "http://schema.org",
    "@type": "Movie",
    "@id": "http://www.filmweb.no/film/article1139116.ece",
    "name": "Jungelboken (The Jungle Book) - 2016 - Filmweb",
    "url": "http://www.filmweb.no/film/article1139116.ece",
    "datePublished": "2016-04-15",
    "genre": [
        "Action",
        "Eventyr"
    ],
    "image": "http://www.filmweb.no/incoming/article1270554.ece/representations/b/Jungelboken",
    "typicalAgeRange": "9-",
    "director": <PERSON TYPE LIST>,
    "author": <PERSON TYPE LIST>,
    "musicBy": <PERSON TYPE LIST>,
    "inLanguage": "engelsk",
    "description": [
        "Bli med inn i en magisk og frodig verden",
        ...,
        <Description N>
    ],
    "productionCompany": <ORGANIZATION TYPE LIST>,
    "alternateName": ["The Jungle Book"],
    "actor": <PERSON TYPE LIST>,
    "duration": "PT1H45M",
    "trailer": <VIDEOOBJECT TYPE LIST>,
    "review": <REVIEW TYPE LIST>,
    "countryOfOrigin": {
        "@type": "Country",
        "name": "USA"
    },
    "aggregateRating": {
        "@type": "AggregateRating",
        "bestRating": "10",
        "ratingValue": "7.1",
        "ratingCount": "2940"
    }
}
```

Listing 6.15: Movie JSON-LD (minimized)

Chapter 7

Survey

This chapter will describe the surveys performed to gather user data and ratings. Google Forms [1] were used for both surveys.

7.1 First Survey

This survey consists of 28 parts, in Norwegian and is distributed to friends and family via Facebook [2].

Part 1 An introductory part where users leave gender and age. This to know the variation of the users.

Part 2 Three questions, first one how often the user watches movies on the cinema yearly, to get a wider user profile. Second asks the user to list movies watched on the cinema the last 12 months, to get more rated movies. And the last question asks to list favorite movies, to get more rated movies.

Part 3 A descriptive part of the next parts.

Part 4 to 21 The main parts of the survey, where the user rates 18 different selected movies. The 18 movies selected was based on screening events in Oslo 16th of May 2016. Due to the massive amount of screenings at that date, the set was further reduced, by inly selecting movies with the Norwegian release date in April or May 2016.

Each part is of one movie each, and contains name, a trailer and movie facts. There was four answer options to the question ”Is this a movie you want to watch on the cinema theater?” This to get rating training data for the recommender models in chapter 8.

[1] `docs.google.com/forms`

[2] `www.facebook.com`

1 Yes, absolutely! I am going to watch it, or has done so already.

2 Yes, I would like that.

3 I don't know.

4 No.

Part 22 This part comprises of questions whether the user find a feature relevant or not. With the first question rating relevance of features as seen in Table 7.1, this is to find out what features that the first basic CB recommender models should use.

The two next questions are about what genres they like and don't like to watch on the cinema, this to see of the weights learned in subsection 8.3.1, somewhat matches what the user have answered.

Part 23 The end of the survey, where responders can leave their mail for a follow up survey.

	Not relevant	Somewhat relevant, but not essential	Somewhat important	Very important
Friends watching the movie:				
Genre:				
Reviews:				
Story:				
Trailers:				
Images:				
Director:				
Music composer:				
Duration:				
Nationality:				
Language:				
User ratings:				
Actors:				
Video Format:				

Table 7.1: How important are the following features for you to watch a particular movie at the cinema?

7.2 Second Survey

This survey is shorter, as it only aims on getting rating for movies on an another date and location. To simulate use of an application, the user have selected Trondheim city area as location filter (see. subsection 2.5.6), and 31st of May 2016 as the temporal filter (see. subsection 2.5.5). The first survey can be thought of as previously rated movies, and those movies should not be shown at all, this leaves us with 13 movies not yet rated.

The aim is to rank the 13 not yet rated movies with the most relevant movie on top of the list. This survey is therefore made to get ratings for the 13 new movies, and use them as mostly test data after the training from the first survey.

There is one question with 13 rows and 4 answers each similar to the part 4 to 21 answers in section 7.1.

Chapter 8

Screening Event Recommendation

This chapter will describe how the different recommendation techniques are performed on the data gathered from the two previous chapters.

8.1 Data Setup

Movie data from chapter 6 is decomposed into vectors representing the different movies. IMDb[1] is used for of previously watched and favorite movies from the survey [chapter 7] that is not already in the dataset from chapter 6.

Based on answers from chapter 7 and resulting statistics from section 9.1, the following data and features is used:

Movie ID The integer value of the ID is chosen to represent each unique movie. Movie ID from IMDb is multiplied by -1 to not get collisions.

Movie Name For a more user friendly readable ranked list, and debugging purposes.

Description xN All movie descriptions in concatenated into one longer text, remove all non alphabetic characters and set all to lower case. Then TF-IDF [subsection 2.6.1] will be used as the feature vectorization giving N features depending on minimum document frequency and total different words.

Genre x28 Each different genre will be represented by an binary index depending on the movie genre compositions. (i.e. Drama, Romance movie = [1,1,0,...,0])

Average User Rating Decimal between 1-10 based on ratings from users on Filmweb.

Average Review Rating Decimal between 1-6, based on reviews.

[1] `www.imdb.com`

A user-item-rating triple is generated from survey answers, both the explicitly questioned movies, and the previously watched and favorites:

User ID Number from 1 to 53.

Movie ID The integer value of the ID is chosen to represent each unique movie. Movie ID from IMDb is multiplied by -1 to not get collisions.

Rating Based on answers from survey.

Rating = 2 For answer ”Yes”, ”Have seen” and if listed in favorite or other watched movies. As all those movies is deemed highly relevant to rank early in a recommendation.

Rating = 1 For answer ”Don't know”. The user don't know and is an indication of a movie that should be ranked over the ”No” answers.

Rating = 0 For answer ”No”, the user have explicitly stated that this is not a move he/she wants to be recommended on top of a recommendations list.

In addition a user-item-rating from review data is also generated, assigning a user id to each individual reviewer and assign the rating from 0-2 as follows: 1 or 2 = 0, 3 or 4 = 1 and 5 or 6 = 2.

8.2 Performance Metrics

RMSE [section 2.8.1] will be used to evaluate how close the predicted values are to the actual ratings, this will give a small indication if one recommendation technique is better than another. But as there are only three different rating, it should be easy only giving a score of 1 to all movies and thus never get more than 1 in RMSE.

As we want a recommender placing the most relevant documents on top of a ranked list, and the least relevant documents on the bottom, a better performance test for ranking is nDCG [subsection 2.8.2]. The different recommenders will therefore also be scored with nDCG@13, nDCG@6 and nDCG@4, that will we used to conclude the best recommendation techniques for the gathered data sets.

In addition to score the different recommenders against themselves, four baselines will also be used:

Wors possible rank For each user, the given predicted score will be the in inverse of the actual score. (i.e. predict rating of 2 to an actual 0)

Random descimal rating The predicted score for each movie is set to a decimal between 0.0 and 1.0, chosen at random.

Random integer rating The predicted score for each movie is set to 0, 1, or 3, chosen at random.

Most Popular Calculate average from other users who have already rated the item.

8.3 Recommendations using Apache Spark

Apache Spark [section 5.3] is used to train models and predict scores for a ranked list of recommendations. Different combinations of data from the first survey and collected data is used as training sets. While data from the second survey is used as test sets.

8.3.1 Content-based Filtering Recommender

Linear regression with SGD [subsection 2.5.7] will be used to train the model of feature weight for each user. The model will be trained giving it a list of labeled points (Pairs of ratings and movie vectors) from the training data from each user individually.

The model will be trained with different feature sets separately, then combining them based on individual performance. There are four sets of features: Extracted description features from TF-IDF, genre features, average user rating and average review rating. And is tested with and without an added bias feature.

Different parameters for will also be tuned to find the best performance. There are three different parameters: Number of iterations, lambda and minimum document frequency for the extracted description features.

8.3.2 Collaborative Filtering Recommender

ALS [section 2.5.8] will be used to train the model, and learn the latent features. The only data needed is the user-item-rating lists.

For scoring and testing purposes, we will cross validate the data, iterating over each user. And use only data from the second survey as test sets on a per user iteration, and the rest as training.

The different training data that will be tested is as follows: Only the asked movies from survey, add the rating from the listed favorites and watched, only asked from survey and reviews as extra users, all the previous combined. There are also four training sets similar to these, but with only the users who have performed both surveys.

Different parameters for will also be tuned to find the best performance. There are three different parameters: Number of iterations, regularization parameter lambda and number of latent factors.

8.3.3 Hybrid Recommender

The final recommender will be a simple wighted hybrid, using a 50-50 distribution from the scores generated by the best CB and CF recommender techniques from the two previous sections.

Part IV

Evaluation

Chapter 9

Results

This chapter will present result from the results gathered during the different parts of the project.

9.1 Web Scraping results

Data for cinema collection and a full scraping of all future screening events available on Filmweb at the 31st of May 2016 are used. As shown in Table 9.1 and Figure 6.1 most cinemas are placed in all of Norway. Table 9.2 shows some key values extracted from the 206 gathered movies. Screening numbers are listed in Table 9.3 and visualized in Figure 9.1.

Locations	300
Cinemas	318
Locations with more than one cinema	9
Maximum cinemas in location	9

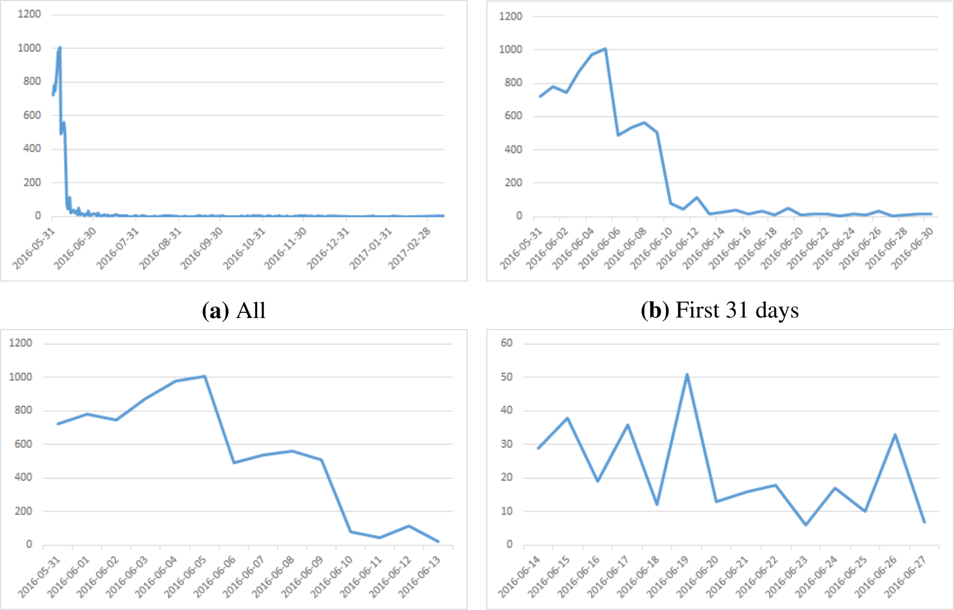
Table 9.1: Statistics gathered cinemas

Movies	206
Maximum genres in one movie	6
Minimum genres in one movie	0
Average genres in one movie	1.39
Actors starring in multiple movies	111
19 Actors starring in multiple movies from 2016	19
Actors starring in multiple movies from April or May 2016	2
Best average user rating	10.0
Worst average user rating	3.0

Table 9.2: Movie statistics from scraped data

Screenings	8165	100%
Screenings first 31 days	7806	95.6%
Screenings first 14 days	7462	91.4%
Screenings next 14 days	305	3.7%

Table 9.3: Screening statistics from scraped data



(a) All

(b) First 31 days

(c) First 14 days

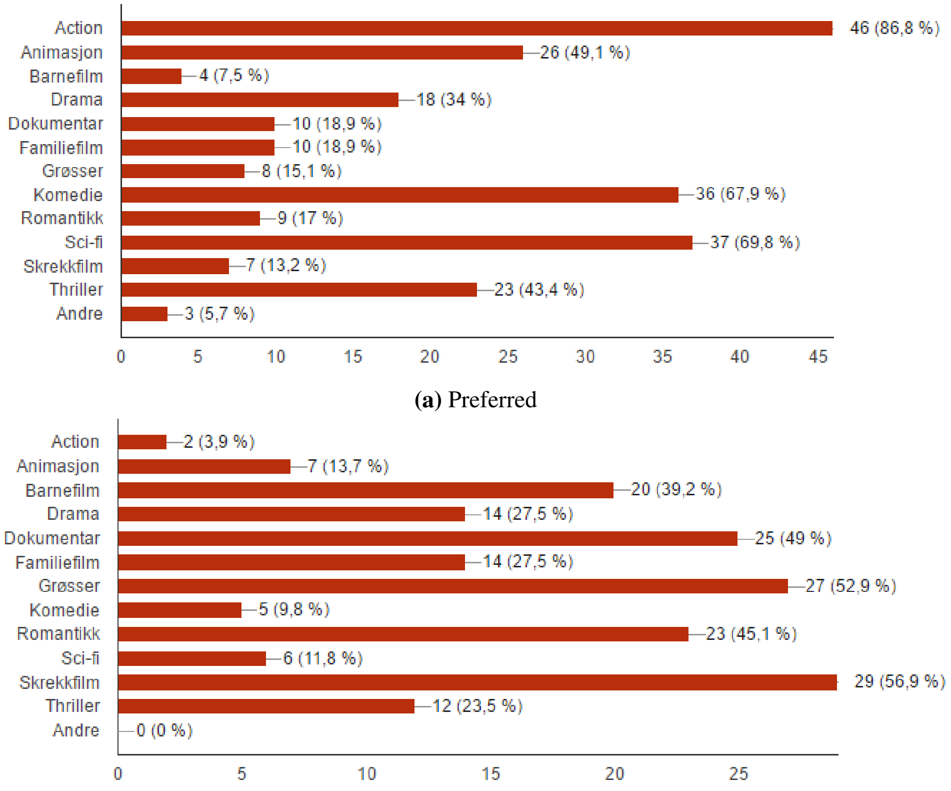
(d) Next 14 days

Figure 9.1: Screening count plots

9.2 Survey results

The first survey was answered by 53 people, where 9 of them answered the follow up survey. Of the 53 users 45.3% were female and 54.7% male, with the age range from 17-53 years old, averaging 24.9 years old. Everyone watched movies at leas once a year with 53.8% attending less then four. 142 additional movies was extracted manually from IMDb based on the answers, and all 18 movies had at least 4 people giving answers resulting in a rating of ”0”, ”1” or ”2”.

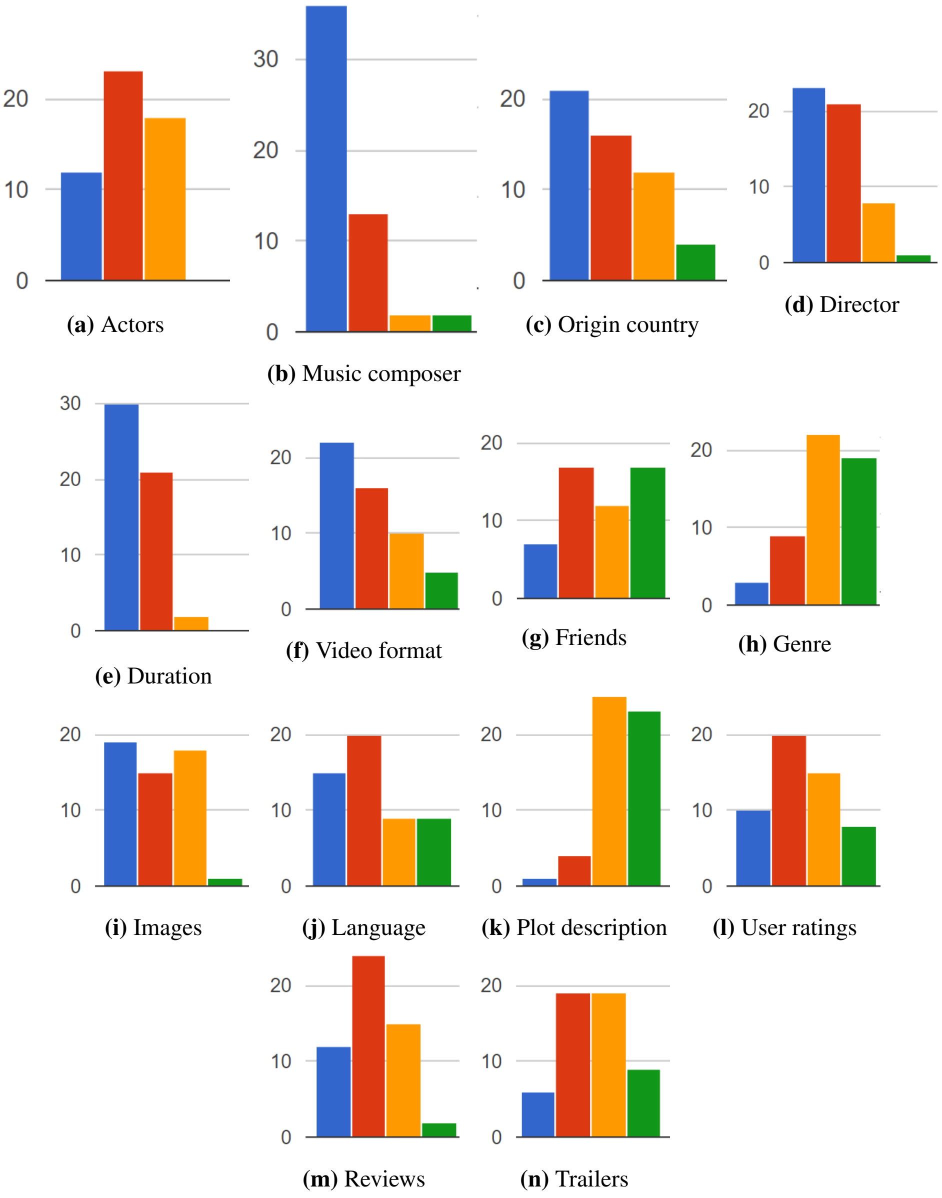
What genres which are most and least preferred, according to the users themselves, are shown in Figure 9.2. And the user feedback rating feature importance is shown in Figure 9.3(blue = ”Not relevant”, red = ”Somewhat relevant, but not essential”, yellow = ”Somewhat important” and green = ”Very important”).



(a) Preferred

(b) Not preferred

Figure 9.2: Genre preferences



(a) Actors

(b) Music composer

(c) Origin country

(d) Director

(e) Duration

(f) Video format

(g) Friends

(h) Genre

(i) Images

(j) Language

(k) Plot description

(l) User ratings

(m) Reviews

(n) Trailers

Figure 9.3: Feature preferences

9.3 Screening Event Recommendation results

This section will present result for the different recommenders. All plots are compared with a random run of a random rank recommender.

9.3.1 Results From Content-based Filtering Recommender

The plots in Figure 9.4 shows the different results for the CB recommendations. Features performing best in combination was genres together with the average user rating from Filmweb. Combined feature experiments performing worse or similar to TF-IDF is not shown.

The learned weights from the best linear regression for each user is shown in Table 9.4.

The experiments are as follows:

Experiment 0 Random

Experiment 1 Reviews

Experiment 2 TF-IDF

Experiment 3 User Rating

Experiment 4 Genres

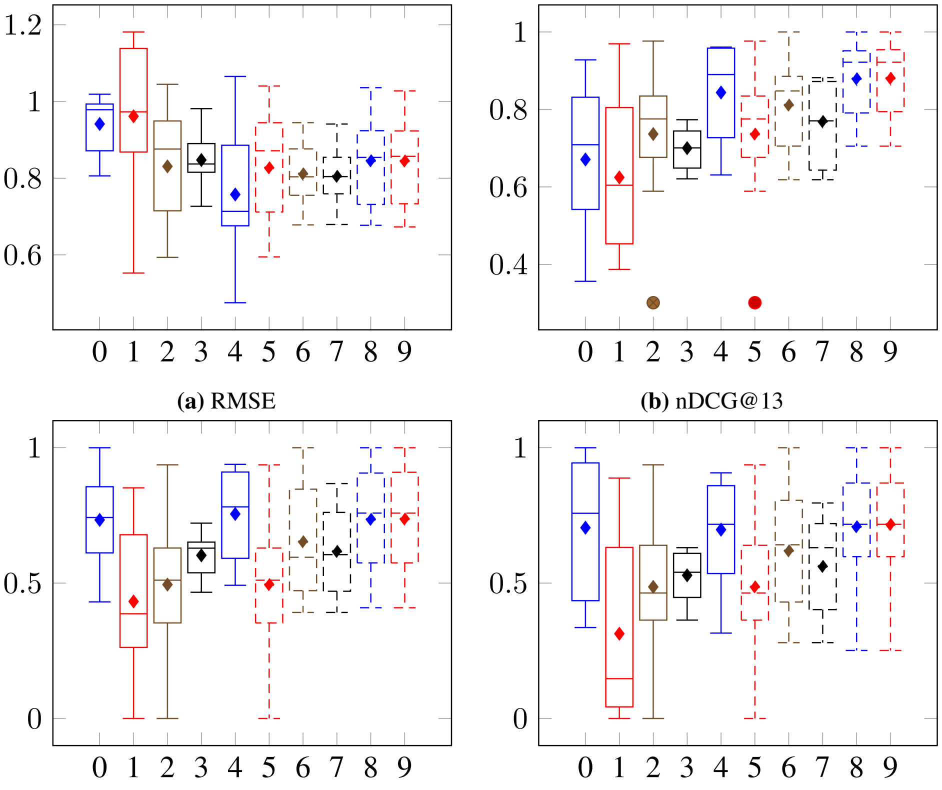
Experiment 5 TF-IDF + Genres,

Experiment 6 TF-IDF + Genres + User Rating

Experiment 7 TF-IDF + Genres + User Rating + Reviews

Experiment 8 Genres + User Rating

Experiment 9 Genres + User Rating + Bias



(a) RMSE

(b) nDCG@13

(c) nDCG@6

(d) nDCG@4

Figure 9.4: Content-based filtering recommender box-plots

User ID	1	5	12	19	21	37	39	45	46
DRAMA	0.19	0.03	-0.04	0.11	0.06	-0.04	0.04	-0.08	0.09
ROMANTIKK	0.05	0.03	0.00	0.00	0.00	0.00	0.00	0.00	0.07
ANIMASJON	-0.01	0.02	-0.14	-0.05	-0.17	0.08	-0.13	0.03	-0.05
BARNEFILM	0.08	-0.02	-0.09	-0.00	-0.11	-0.03	-0.17	0.06	0.01
EVENTYR	0.20	0.24	0.17	0.10	0.25	0.18	0.20	0.28	0.13
FAMILIEFILM	0.08	0.07	-0.09	-0.00	-0.11	-0.03	-0.13	0.06	0.01
ACTION	0.18	0.20	0.21	0.26	0.28	-0.02	0.27	0.40	0.28
BIOGRAFI	0.04	-0.02	0.02	-0.05	-0.07	-0.01	0.02	0.01	0.00
KOMEDIE	0.03	-0.23	-0.10	-0.05	-0.03	0.07	0.10	-0.18	-0.22
KRIM	0.02	0.10	0.25	0.11	0.00	0.00	0.00	0.00	0.05
FANTASY	0.05	0.14	0.12	0.00	0.20	0.00	0.00	0.14	0.11
THRILLER	-0.06	-0.11	0.08	0.06	-0.00	-0.01	0.12	-0.02	0.06
EKSPERIMENTELL	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
SCI-FI	0.14	-0.01	0.23	0.21	0.23	-0.11	0.14	0.15	0.23
KRIGSFILM	0.00	0.03	0.00	0.00	0.00	0.00	0.00	0.00	0.00
WESTERN	0.00	0.03	0.00	0.00	0.00	0.00	0.00	0.00	0.00
SORT KOMEDIE	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
SKREKKFILM	-0.01	-0.05	-0.05	-0.04	-0.05	-0.09	-0.08	-0.06	-0.06
DOKUMENTAR	-0.01	-0.05	-0.05	-0.04	0.09	0.00	-0.08	0.01	-0.06
MUSIKKFILM	0.04	-0.05	-0.05	-0.05	-0.07	-0.01	-0.02	0.01	0.00
TAMIL	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
GRØSSER	-0.10	-0.09	-0.09	-0.08	-0.11	-0.19	-0.16	-0.13	-0.04
EROTIKK	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
SPENNING	-0.11	-0.05	-0.05	-0.05	0.01	0.10	0.09	0.09	0.00
KORTFILM	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
MUSIKAL	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.07
FILM-NOIR	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
GANGSTERFILM	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
User Rating	0.17	0.14	0.10	0.07	0.11	0.15	0.16	0.12	0.11

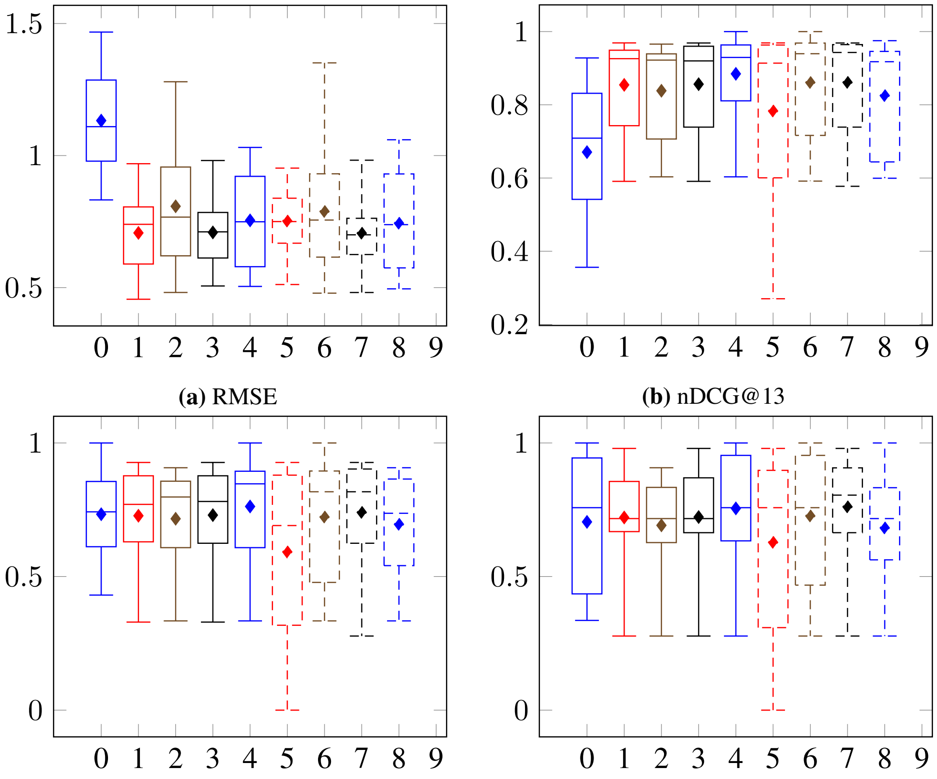
Table 9.4: Learned weights per user, genres and average user rating from Filmweb.

9.3.2 Results From Collaborative Filtering Recommender

The plots in Figure 9.5 shows the different results for the CF recommendations. Using only the 9 test users for training performed best, as it does not need to predict wight for all users. Therefore the experiment using all users, review-users and ratings will be used in further experiments as well.

The experiments are as follows:

Experiment 0 Random
Experiment 1 18 survey movies 53 users.
Experiment 2 18 survey movies 9 users.
Experiment 3 All rated movies 53 users.
Experiment 4 All rated movies 9 users.
Experiment 5 18 survey movies 53 users and review users.
Experiment 6 18 survey movies 9 users and review users.
Experiment 7 All rated movies 53 users and review users.
Experiment 8 All rated movies 9 users and review users.



(a) RMSE

(b) nDCG@13

(c) nDCG@6

(d) nDCG@4

Figure 9.5: Collaborative filtering recommender box-plots

9.3.3 Results From Hybrid Recommender

The plots in Figure 9.6 shows the different results for the hybrid recommendations. Compared to the best individual recommender techniques. The ranking performance from the hybrids have better median and average results.

The experiments are as follows:

Experiment 0 Random

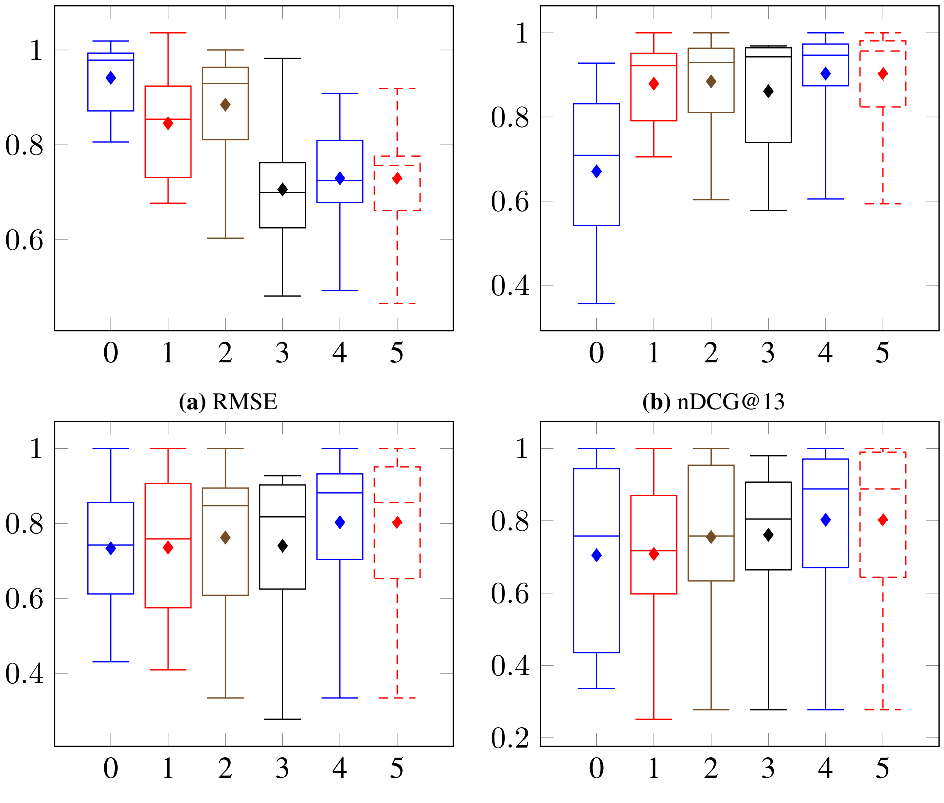
Experiment 1 Genres + User Rating

Experiment 2 All rated movies 9 users.

Experiment 3 All rated movies 53 users and review users.

Experiment 4 Hybrid 1 and 2.

Experiment 5 Hybrid 1 and 3.



(a) RMSE

(b) nDCG@13

(c) nDCG@6

(d) nDCG@4

Figure 9.6: Hybrid recommender box-plots

9.4 Compare to baselines

The plots in Figure 9.7 shows the different results for the hybrid recommendations, compared to the baseline recommendations. Results from the ranked recommendations show that the median and average performance for the hybrid is better then simple baselines.

The experiments are as follows:

Experiment 0 Worst Possible ranking

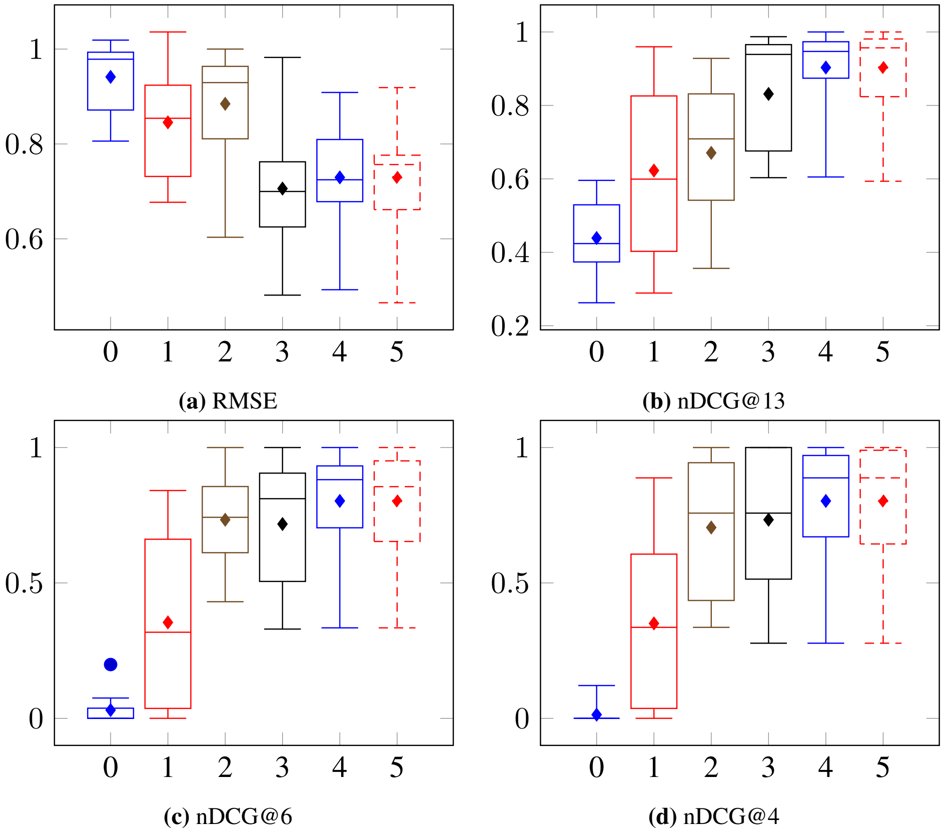
Experiment 1 Random decimal

Experiment 2 Random integer

Experiment 3 Most Popular

Experiment 4 Hybrid 1 and 2.

Experiment 5 Hybrid 1 and 3.



(a) RMSE

(b) nDCG@13

(c) nDCG@6

(d) nDCG@4

Figure 9.7: Best recommenders compared to baseline box-plots

Chapter 10

Discussion

This chapter will discuss the contributions from this thesis and the results presented in chapter 9.

10.1 Web scraper

Construction of a web scraper is tedious work, as most web pages are designed with only user experience in mind. Most content in web sites is therefore usually automatically generated from templates, creating a HTML structure not easily understood or parsed. Edge cases might occur when the web site is not always consistent with the data formatting presented, and are often not detected until later stages.

JavaScript rendered pages are also a problem for web scraping, as you need to run a JavaScript engine such as PhantomJS [section 5.2]. There is also an issue of run time and making sure the script has loaded.`www.example.com` had to be rendered between each request to Filmweb, to make sure the newly requested script would be loaded.

I would therefore recommend for future researchers into this topic to use sources with APIs or get a ”back door” into the data. Mail correspondence with Filmweb was conducted, and they were interested in cooperation, but unfortunately the developers did not have the time.

10.2 Data models

Gathering data and generating appropriate models in a well documented vocabulary made them much easier to work with when they were to be analyzed. There is still difficult to format some of the fields into the appropriate data formate, such as language with no reference to the used format on Filmweb.

10.3 Data Retrieved

Data retrieved show that gathered movies in April and May (Table 9.2) have very few people objects in common, which is logical as actors or producers etc. usually don't star in two different movies released within a short time frame. This shows that there would not be possible to base any of the recommendations on these object types.

Results presented in Table 9.3 and Figure 9.1 shows that the lifespan of the screening events retrieved are at maximum 273, but 95.6% of all screenings are within the next 31 days, and for the next 14 days it is 91.4%. This shows that most screening events are not planned far ahead of time, usually making it unnecessary to perform personalized recommendations[1] when users want to know what is happening in two or more weeks time, as most events don't ”live” that long.

10.4 Surveys

The gathered data from the first survey provided an indication of relevant features for screening event recommendations. Resulting answers (see: Figure 9.3) shows that there are three features in particular that are important, namely: Genre, plot and whether friends are attending as well. Talking to some of the users, it was stated that with this survey they did discover a number of interesting cinematic movies they did not know about beforehand. It was also stated that a good trailer was of some importance, but since it is based on the plot and genre, a recommender would in theory not need to analyze the trailer.

From Figure 9.2a we can deduce the most popular movie genres which are ”action”, ”sci-fi” and ”comedy”. ”Animation”, ”Thriller” and ”Drama” is alls highly preferred, but looking at Figure 9.2b these genres as much disliked. The movies people don't like watching at cinemas are ”Romance”, ”Drama”, ”Documentary”, ”Movies for children” and ”Horror”[2].

The initial plan for the first survey was to gather all data, and split into test and training, but was later decided that there should be performed a seconds survey to simulate the users wanting recommendations on a new date. The problem with this though was that the user poll for the second survey is only nine users, compared tho the 53 people from the first survey. The nine users still had a good diversity of interests and ratings.

10.5 Recommender Discussion

Results for the hybrid recommender, which were based on the two best results from the CB and CF approach, shows the most accurate rated results. The second best was the CF approach, followed on the CB. All preforming better than worst possible, most popular, and most of the random recommenders.

Plots shown in Figure 9.4 shows that the CB approach did not perform much better than random on nDCG@6 or nDCG@4, but keep in mind this was a particularly good random run of the random ranking and the CB approach gives consistent results. There

[1] When applying a temporal and location filter

[2] The Norwegian genre terms ”Skrekkfilm” and ”Grøsser” are both translated to ”Horror movie”

best resulting model was trained and tested using only genre and user rating. Many of the retrieved fields was not tested or reasoned about in the recommender model, due to none or few of the field values being present in multiple data models (i.e. only two shared actors between movies.)

The learned weight from the best CB recommender is shown in Table 9.4, some of the weights ar zero for users since they were false in all the trained movies. By comparing learned weights to the genre preferences given in survey one, we can spot some differences where the user have stated that he/she likes or dislikes the genre, but they are weighted in the opposite direction. Example: User 1 has stated that she likes animation movies but it is weighted ”-0.01”. Either the user don’t really like animation movies or the more likely; The model need more data for training, and the rated movies was by chance an uninteresting animation movie. Some preferences can not be modeled by linear regression, such as: A user dislikes plain comedy and sci-fi movies, but loves comedy-sci-fi.

10.6 Overall Discussion

The surveys did not simulate the usual main problem for future event recommendation, which is data sparsity. User cold-start problem was not accounted for in the CB recommender and item cold-start was not used in the CF recommender. This led to good results for the weighted hybrid approach. Showing better then random result for the CB approach with few rated items, there should be implemented a weighted hybrid, starting with higher wights for the CB approach and gradually decrease is and add more weight to the CF model.

For a cold-user situation, there should be implemented an approach using the user profile data such as age gender interests. First start to recommend based other users with similar fields, and gradually shifting to the general CB, CB or hybrid approach (which ever is used).

Chapter 11

Conclusion

This chapter will state the contributions given in section 1.4 and conclude the questions stated in section 1.3

11.1 Research Contributions Review

This section will review the contributions stated in section 1.4.

C1 *Creating a web scraper for collection of screening events, corresponding to RQ1 and the first part of RQ2.*

Conclusion A web scraper of Filmweb for collection future screening events was created. The scraper places almost all data from Filmweb into a JSON-LD model with the Schema.org vocabulary.

Most Norwegian cinema locations were gathered getting the names from a list on Filmweb and manually tagging based on Google maps and Facebook pages or communal homepage for the cinemas.

C2 *Making a survey of cinema movie interest and evaluate what features people themselves deemed important (RQ2). There is also a follow up to gather test data for RQ3.*

Conclusion Two surveys were conducted where the first one had an attendance of 53 users and the second with 9. The amount of users ar not statistically significant, but is used as an indication for future work.

C3 *Evaluate different recommendation techniques in accordance with RQ3. And to evaluate relevance of obtained features as in RQ2.*

Conclusion The data collected was evaluated using built in machine learning methods from Apache spark MLLib. For content-based recommendations a Linear Regression with Stochastic Gradient Descent was used, with a built in TF-IDF model for feature extraction from descriptions. For the collaborative filtering recommendation a Matrix Factorization technique based on Alternating Least Squares were used. The hybrid recommender approach used predicted scores for the two other techniques, with a 50-50 weighting ranking the movies.

11.2 Research Question Conclusions

This section will review to what degree the questions asked in section 1.3 was answered.

RQ1 *How can we in real-time identify the location, date and time of all relevant movies on Norwegian cinemas?*

Conclusion Filmweb is a Norwegian web page containing almost all relevant Norwegian cinemas an their screenings, most if the events are available approximately 10 days ahead of the screening. While further into the future there are generally only available events for movie premieres or the small traveling screenings delivered by Bygdekinoen. They screen at locations used for more than cinema and have screenings less frequently, thus need to plan ahead to have the location available and to get visitor.

Only cinema names and locations are available on Filmweb, but they are static and the data was collected manually. Even with manually annotation of the document, they are not 100% correct as there is no singular place containing addresses or location data for many of the smaller cinemas.

Using the web scraper once a day is one way to gather all the planned screenings at any given time. There is also possible to get planned events based on a particular location or cinema in real time.

RQ2 *Which features can we automatically extract for these movies, and to what extent are these features relevant for movie recommendations?*

Conclusion We can extract a wide variety of features shown in chapter 6. Based on surveys and data analysis the most important extracted features collected was genre and user ratings. If friends of a user would like to attend the event, were one seemingly important feature which could not be collected.

The movie plot/description and trailers is also important features which is harder to analyze, and there is no conclusion of to what extent they are needed for accurate recommendations.

RQ3 *How do collaborative filtering, content-based recommendations and hybrid recommendation strategies compare with the features retrieved?*

Conclusion Based on surveys and data analysis setup using genre and user rating gives better predictions than the baselines in content-based recommendations. With only 9 users with varying tastes, the collaborative filtering recommendations performed even better. And the 50-50 weighted hybrid recommender conducted the best performance on average.

The reader should also keep in mind that filtering based on temporal and location data is a binary preliminary filter before the recommender techniques is applied. Resulting in very few screening events at a given time and place, which reduces the need of a perfectly accurate recommender to give the user information about the most relevant future screening events. There would be a greater need of a good recommender if the filters were to be turned off, leading to ranking 206 movies.

Chapter 12

Future Work

This chapter will outline potential future work affiliated with future event recommendation.

12.1 Other Machine Learning Approaches

This thesis only used linear regression, Matrix Factorization and a simple wighted hybrid. Future research should look into other approaches, which might suite the domain better.

12.2 Recommend based on more data

The data collected for this thesis is rather small, and there was no good way to recommend based on sparsity, as there was little sparsity. Using the simple recommender techniques presented in this paper, events could be implemented in an application and use gathered data to improve and do research into potentially better methods.

If we were to recommend based on actors, producers, directors, etc. we need data form a much larger time period as discussed in section 10.3.

12.3 Cold start Approaches

Earlier work performing studies on future event recommendation states that the big problem is the item cold-start problem. With the small sample set collected from the surveys in this project there was not conducted analyzes of cold-start events. With more item and user data, more research should be conducted into how to recommend cold items.

12.4 Other future event types

Contribution from this thesis only focuses on a particular event type. Future research should try to incorporate larger data sets with a variety of event types. The final recommender could consist of separate recommenders for each event type and use the results as input for a final recommender modeling the user preference for each event type.

12.5 Automatically detect event reviews in media

As this thesis is a part of the SmarMedia program [subsection 1.2.1], it would be interesting to retrieve reviews from the news automatically. This is due to many future events are not linked to their reviews like they usually are on Filmweb.

Bibliography

Burke, R., 2002. Hybrid recommender systems: Survey and experiments†. User Modeling and User-Adapted Interaction.

Cornelis, C., Guo, X., Lu, J., Zhang, G., 2005. A fuzzy relational approach to event recommendation. IICAI '05.

de Macedo, A. Q., Marinho, L. B., 2014. Event recommendation in event-based social networks. SP '14.

Dooms, S., Pessemier, T. D., Martens, L., 2011. A user-centric evaluation of recommender algorithms for an event recommendation system. RecSys '11.

Ji, X., Xu, M., Zhang, P., Zhou, C., Qiao, Z., Guo, L., 2015. Online event recommendation for event-based social networks. WWW 2015.

Kayaalp, M., Õzyer, T., Õsyer, S. T., 2009. A collaborative and content based event recommendation system integrated with data collection scrapers and services at a social networking site. ASONAM '09.

Khrouf, H., Troncy, R., 2013. Hybrid event recommendation using linked data and user diversity. RecSys '13.

Koren, Y., Bell, R., Volinsky, C., 2009. Matrix factorization techniques for recommender systems. Computer.

Liu, X., He, Q., Tian, Y., Lee, W.-C., McPherson, J., Han, J., 2012. Event-based social networks: Linking the online and offline socialworlds. KDD '12.

Macedo, A. Q., Marinho, L. B., Santos, R. L. T., 2015. Context-aware event recommendation in eventbased social networks. RecSys '15.

Minkov, E., Charrow, B., Ledlie, J., Teller, S., Jaakkola, T., 2010. Collaborative future event recommendation. CIKM '10.

Purushotham, S., Kuo, C.-C. J., 2015. Modeling group dynamics for personalized group-event recommendation. SBP '15, 405–411.

Qiao, Z., Zhang, P., Cao, Y., Zhou, C., Guo, L., Fang, B., 2014. Combining heterogenous social and geographical information for event recommendation. AAAI '14.

Rendle, S., Freudhaler, C., Gantner, Z., Schmidt-Thieme, L., 2009. Bpr: Bayesian personalized ranking from implicit feedback. UAI '09.

Stern, D., Herbrich, R., Graepel, T., 2009. Matchbox: Large scale online bayesian recommendations. Proceedings of the 18th International World Wide Web Conference.

Zhang, Y., Wu, H., Sorathia, V., Prasanna, V. K., 2013. Event recommendation in social networks with linked data enablement. ICEIS2013.