

Towards a Social Web based solution to bootstrap new domains in cross-domain recommendations

Master's Thesis

Martijn Rentmeester

Towards a Social Web based solution to bootstrap new domains in cross-domain recommendations

THESIS

submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE

in

COMPUTER SCIENCE

by

Martijn Rentmeester
born in Goes, the Netherlands



Web Information Systems
Department of Software Technology
Faculty EEMCS, Delft University of Technology
Delft, the Netherlands
<http://wis.ewi.tudelft.nl>

Towards a Social Web based solution to bootstrap new domains in cross-domain recommendations

Author: Martijn Rentmeester
Student id: 1308246
Email: martijnrentmeester@gmail.com

Abstract

Most recommender systems recommend items from a single domain. However, usually users' preferences span across multiple domains. Cross-domain recommender systems can successfully recommend items in multiple domains when there is knowledge about the user's preferences for items in at least one of the domains and when there is knowledge about relationships between domains. But when a new domain is added to a cross-domain recommender system, this knowledge usually lacks and giving cross-domain recommendations is not a trivial problem anymore. Current approaches use content-based relations to bootstrap new domains in cross-domain recommendations. In this thesis we propose a new model that transfers existing users' preference based relations between domains from an auxiliary Social Web system to a cross-domain recommender system in which a new domain needs to be bootstrapped. In a case study on the Open Images dataset we researched this solution to get insight in how well the model works and whether it has potential for widespread usage.

Thesis Committee:

Chair: Prof. dr. ir. G.J.P.M. Houben, Faculty EEMCS, Delft University of Technology
University supervisor: Dr. A. Bozzon, Faculty EEMCS, Delft University of Technology
Committee Member: Dr. K.V. Hindriks, Faculty EEMCS, Delft University of Technology

Preface

This thesis is the end result of my Master's thesis project, which I performed at the Web Information Systems (WIS) group at Delft University of Technology. It is the final product of my study that started smoothly eight years ago. After finishing my Bachelor's degree in three years I decided that it was time to develop myself in other areas as well and became an active member of the mathematics and computer science study association and took a part time job. I enjoyed this time, until I realized at some point in time that I was ready for a new challenge and that I should finish my study.

I decided to start graduating at another department than in which I took my Master's courses as my interest in computer science areas had changed over time. Especially in the beginning of my thesis this created some difficult moments which helped to develop me personally. It was also the first time during my study period that I had to work hard, which was good for me. It might be that the quality of this thesis is lower than when I would have created it in the department in which I took my courses. But I take that for granted as I'm glad that I took the decision to challenge myself and learned a lot about Information Retrieval and Web Information Systems during my thesis.

Before I close the final chapter of my study period I would like to thank some persons that helped me creating this thesis. First of all I would like to thank my supervisors Alessandro Bozzon and Jasper Oosterman for guiding me and challenging me with good questions during our weekly meetings. Second, I would like to thank Claudia Hauff for giving feedback during the two-weekly meetings with all graduate students. Furthermore I would like to thank Geert-Jan Houben and Koen Hindriks for being part of my thesis committee and providing me with feedback.

During my research I discussed with many other persons and as such I would like to thank all the people that helped me with this thesis project. To conclude this preface I would like to thank my family, girlfriend and friends for supporting me during this thesis and my university career.

Martijn Rentmeester
Delft, the Netherlands
August 14, 2014

Contents

Preface	iii
Contents	v
List of Figures	vii
1 Introduction	1
1.1 Cross-domain recommendations	2
1.2 Cold-start problems	3
1.3 Research objectives	4
1.4 Contributions	6
1.5 Outline	6
2 Background	7
2.1 Definition of domain	7
2.2 Recommender systems	7
2.3 Cross-domain recommender systems	12
2.4 Cold-start problem	14
2.5 Evaluating recommender systems	16
3 A Social Web based solution to bootstrap new domains	19
3.1 Discussion of existing approaches	19
3.2 Introducing our new model	21
3.3 Summary	23
4 Case study: dataset	25
4.1 Requirements	25
4.2 Open Images dataset	25
4.3 Domain taxonomy for the Open Images dataset	26
4.4 Domain annotation task	29
4.5 Summary	33
5 Case study: implementation	35
5.1 Implementation of the new introduced model	35

5.2	Tuning the model parameters in an offline experiment	39
5.3	Discussion of the results of the offline experiment	42
6	Case study: user study and results	45
6.1	Experimental setup	45
6.2	Results of the user study	50
7	Conclusions and Future Work	57
7.1	Conclusions	57
7.2	Discussion/Reflection	59
7.3	Future work	60
	Bibliography	61
A	Glossary	67
B	Metadata Open Images videos	69
C	Metadata YouTube videos	73
D	Results YouTube Experiment	79
E	Results Offline Experiment	83
F	Results majority voting User Study	95
G	Comments User Study	97
H	Recommendations for video documentary/report 5	99

List of Figures

1.1	The focus of this work in the context of recommender systems	5
2.1	The working of the two most used approaches in recommender systems .	8
2.2	A cross-domain recommendation using collaborative filtering in a situa- tion of overlap	13
3.1	The new proposed model	22
4.1	An overview of the annotation task	31
4.2	The level of agreement	33
5.1	Our model applied on the Open Images dataset with the chosen auxiliary system using the domains news item and documentary/report as example .	36
5.2	The Lucene pipeline to create TF-IDF vectors from an unstructured text .	37
6.1	Conceptual model	46
6.2	Overview of the pairwise comparison task	48
E.1	Average Precision@5 for the number of search results for the strategy Title	83
E.2	Average Precision@5 for the number of related videos for the strategy Title	84
E.3	Average Precision@5 for the number of similar Open Images videos for a given YouTube video for the strategy Title	85
E.4	Average Precision@5 for the different scoring functions for the strategy Title	86
E.5	Average Precision@5 for the number of search results for the strategy Title & Date	87
E.6	Average Precision@5 for the number of related videos for the strategy Title & Date	88
E.7	Average Precision@5 for the number of similar Open Images videos for a given YouTube video for the strategy Title & Date	89
E.8	Average Precision@5 for the different scoring functions for the strategy Title & Date	90
E.9	Average Precision@5 for the number of search results for the strategy Title & Domain	91

E.10 Average Precision@5 for the number of related videos for the strategy Title & Domain	92
E.11 Average Precision@5 for the number of similar Open Images videos for a given YouTube video for the strategy Title & Domain	93
E.12 Average Precision@5 for the different scoring functions for the strategy Title & Domain	94

Chapter 1

Introduction

The amount of digital information has grown enormously over the last decades. This has resulted in an information overload. Recommender systems have been introduced to help humans in information seeking tasks cope with this information overload. Recommender systems are systems that produces personalized recommendations as output or has the effect of guiding the user in a personalized way to interesting or useful items in a large space of possible options. The discipline of recommender systems studies methods and algorithms to improve recommender systems and is a sub-field of information retrieval. But differently from standard information retrieval systems, recommender systems retrieve relevant items from a content collection without requiring the user to explicitly express her information need by a search query. Instead it uses a user profile, containing the user's preferences, to filter relevant items and recommend those to the user. Well-known examples of recommender systems are Amazon, Netflix, MovieLens and Last.fm.

Currently, most recommender systems only recommend items belonging to a single domain¹, e.g. Netflix only recommends videos and Last.fm only recommends music. However, human preferences may span across multiple domains [49]. Cross-domain recommender systems are a class of recommender systems that specializes in recommending items across multiple domains. An example of a cross-domain recommender system is the e-commerce website Amazon.com. On this website users are recommended e.g. DVDs, music, books and video games.

Although the research field of cross-domain recommender systems got more attention recently, problems related to cross-domain recommender systems still exist. One of these problems, known as "cold-start problem", arises when a new domain is added to the recommender system or when a complete new system is set up. In those situations there is not enough information about the users' preferences in the newly added domain to come up with good recommendations. Lets consider as an example a

¹A domain is a set of items that share a certain characteristic. This characteristic can for example be:

1. A shared item type, e.g. CDs, books or movies;
2. A shared representation of users' preferences, expressed in explicit feedback or implicit feedback;
3. A shared timestamp;
4. A shared video type, e.g. news items or documentaries.

As the definition using a shared item type is the most straightforward we will use this definition in the examples.

movie recommender system and a user that prefers the movie "Harry Potter". To this system, a new domain of CDs is added. Because the user's preferences towards CDs is unknown, the system has no explicit information to use for recommending CDs.

Cold-start cross-domain recommendations can be computed in two ways, content-based and using collaborative filtering. Current approaches use content-based methods to relate items from the new domain to items from the existing domains, for instance using Semantic Web based solutions or tags. As it is known that collaborative filtering is the best method to compute recommendations in a single domain we decided to explore collaborative filtering (read: users' preferences) to relate items from the new domain to items from the existing domains. However a constraint of the situation that we research is that there is no previous knowledge of users' preferences. That makes it impossible to do standard collaborative filtering and we should come up with a new solution to solve this complex problem as to the best of our knowledge this problem has never been researched before.

Therefore, in this thesis, we propose a new model to solve the problem of bootstrapping new domains in cross-domain recommender systems without previous knowledge of users' preferences. The new model uses an external source to gather the missing users' preferences. It uses the Social Web for this purpose. The Social Web is a set of social relations that link people through the World Wide Web [4] and some of the Social Web systems, e.g. YouTube, contain users' preferences. Such a Social Web system will be used as an auxiliary system to relate the new domain to the existing domains based on users' preferences. If in this auxiliary system there is knowledge about users' preferences in the new and existing domains, the model transfers this knowledge to our system. Next, this knowledge is used in our system to relate items from the new domain to items from the existing domains such that items from the new domain can be recommended to a user that prefers an item in one of the existing domains. The advantage is that this model uses already existing knowledge such that a new domain can immediately be bootstrapped. Therefore it is no solution to just upload items from the new domain to an auxiliary Social Web system such as YouTube as it will take time before users' preferences towards these items will be known.

We researched the proposed solution in a case study on the dataset of Open Images², using YouTube as the auxiliary system from the Social Web, to get insight in how well the model works on this dataset and to see whether the model has potential for widespread usage.

In this introductory chapter the subjects of this thesis, cross-domain recommendations and cold-start problems, are discussed more in-depth in section 1.1 and 1.2. Next, the research objectives and contributions are described in section 1.3 and section 1.4. Finally, the outline of the remainder of this thesis is given in section 1.5.

1.1 Cross-domain recommendations

Two tasks of cross-domain recommender systems are defined in literature. The first task is to transfer and exploit user knowledge acquired in one domain into several other domains. This can help a user by recommending her with items in a novel, unexplored domain. This is possible because there might be dependencies between preferences in

²<http://www.openbeelden.nl/>

different domains. The second task is to recommend items from multiple domains in a joint recommendation. This can lead to more diverse recommended items. The first task is in general seen as the true cross-domain recommendations task and is the task researched in this thesis.

To be able to recommend a user items in a novel, unexplored domain, relations between the new domain and the existing domains have to be known, as well as user's preferences from at least one of the existing domains. Cremonesi et al. [10] differentiates four different situations of overlap (relations) between domains:

1. **No overlap:** There is no overlap in items and users between domains. This situation arises e.g. when a new domain is added to a recommender system or when a complete new system is set up;
2. **User overlap:** There are some users that have expressed preferences in both domains. For example one user has expressed preferences in a certain book and a certain movie;
3. **Item overlap:** There are some items in which preferences has been expressed by users from both domains. For example two content providers share a catalog of items (e.g. two TV providers broadcasting the same set of TV channels);
4. **Full overlap:** There is overlap in items and users between domains.

Most research addressed the scenario where some overlap is available. Collaborative filtering is the most used technique to solve that scenario. How this works is explained in the following example. Let us consider a cross-domain recommender system containing the domains books and movies and a user that prefers the book "Harry Potter". This user wants the system to recommend movies to her. A collaborative filtering technique will look for users that also like the book "Harry Potter" and look for their movie preferences. Assuming one of such user expressed a preference for "Lord of the Rings", then collaborative filtering assumes that the user that wants recommendations from the domain movies might also like the movie "Lord of the Rings" and recommends it to her.

Such a setting is not suitable for a no overlap situation. Fernández-Tobías et al. [14] emphasizes in their survey that approaches have to be developed that find or build some type of explicit/implicit relations between domains, to address the no overlap situation. Some approaches have been proposed already. Those approaches relate domains content-based. The drawback of relating domains content-based is e.g. that items that do not share content-based features can not be related to each other. Therefore we propose a new solution in this thesis that relates domains based on users' preferences.

1.2 Cold-start problems

A cold-start problem arises when a new user, item or domain is added to a recommender system, creating a situation in which not enough information about users' preferences for items is available to come up with good recommendations. This situation can also occur when a complete new system is set up. These different scenarios are discussed below:

1. **New user:** When a new user enters a recommender system, not enough information about her preferences is known to come up with good recommendations. This is a problem that is relevant in both content-based recommender systems and collaborative filtering recommender systems;
2. **New item:** When a new item is added to a recommender system, no one has expressed preferences in that particular item yet. This is especially a problem in collaborative filtering recommender systems;
3. **New domain:** This is a problem in cross-domain recommender systems. It is a form of the new item problem, since actually it is nothing more than adding multiple new items to a system. However, these new added items are from another domain, making it in general harder to relate them to items already part of the recommender system, as, for instance, different domains are expressed in different vocabularies;
4. **New system/new community:** This is a combination of the new user and new item problem. This problem can be solved by increased availability of public rating datasets, since these can be used to bootstrap the new system. However the problem is that organizations often keep that data private according to Schafer et al. [39], either for competitive advantage or privacy concerns.

1.3 Research objectives

Before we discuss our research questions, to summarize the previous sections the positioning of our work is visualized in figure 1.1.

Next, we discuss our research questions. As in this thesis we propose a new approach to bootstrap new domains in cross-domain recommendations that makes use of a Social Web system, we pose the following research question:

RQ What is the potential of a Social Web based solution to bootstrap new domains in cross-domain recommendations?

To be able to answer this generic research question, we pose the following questions:

RQ1 Which are the strengths and limitations of the current approaches to bootstrap new domains in cross-domain recommendations?

RQ2 What is the best way to evaluate our proposed model?

RQ3 What is the best configuration of our proposed model?

RQ4 How well does our proposed model work compared to other approaches?

To find an answer to these research questions, we researched the proposed model in a case study.

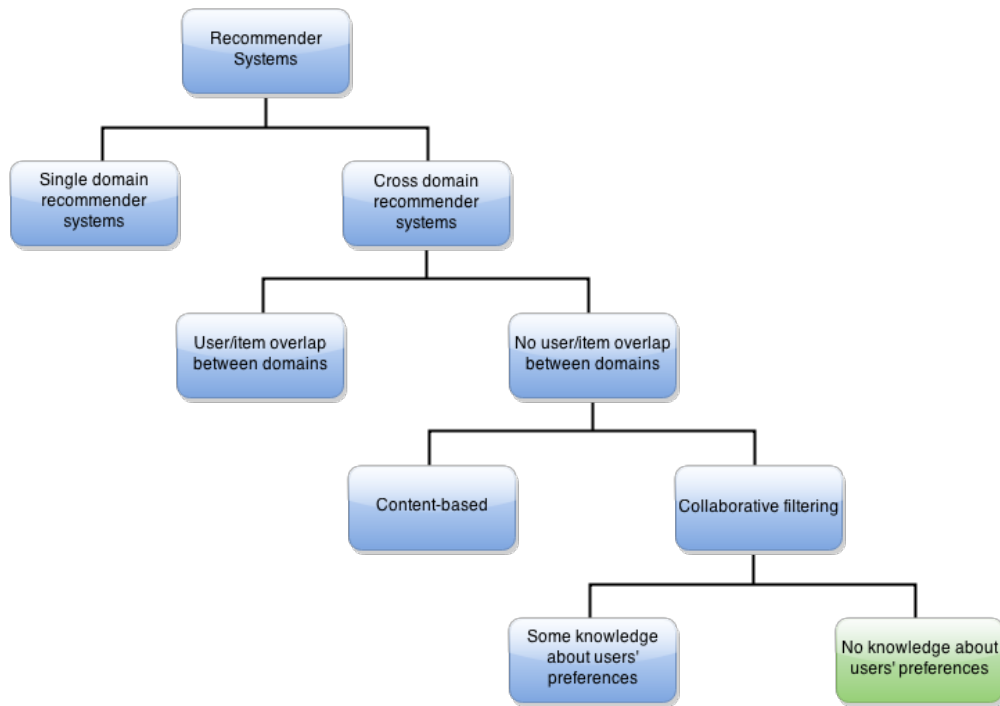


Figure 1.1: The focus of this work in the context of recommender systems

1.3.1 Case-study

In this section the case study is shortly introduced. As reference dataset, we selected the Open Images dataset, which is a freely available dataset from the Sound and Vision institute in the Netherlands. This dataset contains videos about the cultural heritage of the Netherlands. We picked the definition of domain most suitable for this dataset, namely that a domain is a group of items characterized by its video type. Using this definition we annotated the dataset in a manual annotation task. Next, we selected YouTube as auxiliary system from the Social Web as it is the most popular video website. However because YouTube is a black box, it comes with some limitations. Finally, we implemented our model and tuned the parameters of the model in an offline experiment. The results of this offline experiment were used to compare our model to a tag-based and random solution in a user study.

1.3.2 Approach

The following approach was taken to answer the research questions of this thesis. First a literature study was performed to answer **RQ1** and **RQ2**. Next, we proposed our model and selected a dataset to test this model. An offline experiment was set up to tune the parameters of the model on this dataset. The outcomes of the experiment are used to answer **RQ3** and were used to compare the new model to existing approaches in a user study. Our new model was compared to tag-based recommendations and a baseline where videos were randomly recommended. This answered **RQ4**. An analysis was performed to understand the outcomes of the user study so that we could

better understand the potential of the new model.

1.4 Contributions

The contributions made in this thesis are threefold:

1. A model that uses an auxiliary system from the Social Web to cope with bootstrapping new domains in cross-domain recommendations is proposed and evaluated;
2. Using the Open Images dataset as a reference dataset, in a case study we compared a couple of configurations of the proposed model to tag-based recommendations and a baseline where videos were randomly recommended. The outcome of this comparison provides insight in the potential of the proposed model;
3. A new version of the Open Images dataset was created where videos are annotated with domains.

1.5 Outline

The remainder of this thesis is organized as follows. Chapter 2 introduces the definition of domain and contains background information about recommender systems in general, cross-domain recommender systems, the evaluation of recommender systems and solutions to the cold-start problem. Next, chapter 3 discusses the current approaches that can solve the problem of bootstrapping new domains in cross-domain recommendations and introduces our new model. This model is implemented in a case study. The dataset used in this case study is discussed in chapter 4. The implementation of the model is described in chapter 5 together with the offline experiment to tune the parameters of the model. In chapter 6 the user study and the results that followed from this are discussed. Chapter 7 ends this thesis by concluding the work and recommending future work.

Chapter 2

Background

This chapter introduces previous work related to the topics covered by this thesis. It serves two purposes: one, it positions this thesis with respect to existing research, and two, it introduces the set of theoretical and technical tools used in this work.

Before we introduce previous work, first we give a definition of domain in section 2.1. Next a general introduction in recommender systems is presented in section 2.2, and then previous work related to cross-domain recommender systems is presented in section 2.3. Next, works that have tried to recommend items in a cold start situation are discussed in section 2.4. Finally related work to the evaluation of recommender systems is presented in section 2.5.

2.1 Definition of domain

As Fernández-Tobías et al. [14] notice in their survey, there is no consensus on the notion of domain in the literature. For example, Li [24] already distinguishes three different types of domains: system domains, data domains and temporal domains. System domains are the different datasets, e.g. music dataset and book dataset, upon which recommender systems are built. Data domains are the different representations of users' preferences, expressed in explicit feedback (e.g. ratings) or implicit feedback (e.g. visiting history). Finally, temporal domains are subsets in which a dataset is split on timestamps. Because there is no consensus, we define domain as:

Definition. *A domain is the name of a group of items that share a certain characteristic.*

An example of such a characteristic is an item's type. In that case, examples of domains are CDs, books or movies.

2.2 Recommender systems

Recommender systems are systems that produce personalized recommendations as output or that have the effect of guiding the user in a personalized way to interesting or useful items in a large space of possible options [9]. In a typical recommender system the input consists of items and users. Users might express their preferences in

items (in general via ratings) and based on this input the system recommends items to the user. This is clarified in figure 2.1.

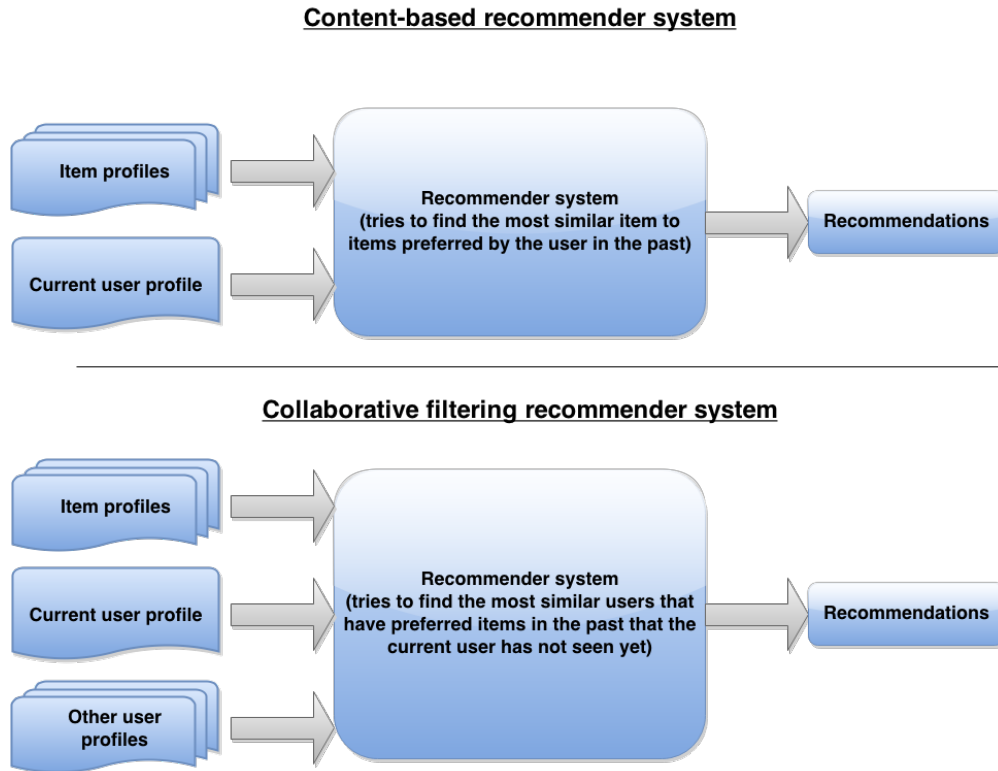


Figure 2.1: The working of the two most used approaches in recommender systems

According to Adomavicius and Tuzhilin [2] the problem of recommendation can be reduced to the problem of estimating ratings (i.e. user's preferences) for the items that the user has not rated yet. Furthermore they state that a recommender system wants to choose an item that maximizes the user's utility. Examples of existing recommender systems are Amazon, Netflix, MovieLens and Last.fm.

The research area of recommender systems became important around the mid 1990s and emerged as a sub-field of information retrieval to help users to deal with information overload. According to [9] the criteria 'individualized' and 'interesting and useful' separate recommender systems from information retrieval systems. Furthermore recommender systems, in contrast to information retrieval systems, filter relevant items from a content collection without requiring the user to explicitly express her information need by a search query. Instead it uses a user profile, containing the user's preferences, to filter relevant items and recommend those to the user.

Recommender systems can be classified in several ways. The remaining part of this section will be divided according to the classification used in [2]. First previous work related to content-based recommender systems will be discussed in section 2.2.1, followed by work related to collaborative filtering approaches in section 2.2.2. Because content-based methods and collaborative filtering both deal with user profiles, a cross-section is written about it (2.2.3). Finally some work related to hybrid recommendation

approaches is presented (2.2.4). Other issues related to recommender systems like explainability, trustworthiness of the results, privacy and security are out of the scope of this thesis and are therefore not discussed here.

2.2.1 Content-based recommender systems

According to [2], content-based recommender systems recommend the user items similar to the ones the user preferred in the past. Examples are The Daily Learner system [7] and Syskill and Webert's content-based recommender system [31]. To give content-based recommendations a profile of the user is build, containing features of the items previously preferred by the user. These features are stored in item profiles.

Item profiles

As noticed by Pazzani and Billsus [32], an item can be represented in different ways. For example, as unstructured text or as numeric features. The only difference is that when an item is represented in an unstructured way, the features have to be extracted. This can be done using information retrieval techniques [28], like stop word removal and stemming.

Although these information retrieval techniques are widely used in content-based recommender systems, they have some drawbacks, resulting from natural language ambiguity [27] [32]. Semantic analysis is a technique to capture the meaning of words to avoid these natural language ambiguity problems. It incorporates either linguistic or domain-specific knowledge using knowledge bases like WordNet or Wikipedia. Degemmis et al. [12] showed that semantic analysis allows learning more accurate profiles and is an improvement on traditional syntactical content-based methods.

User profile

A user profile is a set of relevant features characterizing a user. In general it only contains the user's preferences. In content-based recommender systems, this means it contains features of the items from which is known that the user prefers it. More information about user profiles can be found in section 2.2.3.

Recommendation algorithms

Using item profiles and a user profile, recommendations can be calculated. Adomavicius and Tuzhilin [2] distinguishes two approaches, heuristic-based and model-based.

The most well-known heuristic-based approach is nearest-neighbor. This approach uses similarity measures. The most popular approach for content-based recommendation is a vector cosine-based measure, which is used to calculate a similarity between the user profile and an item profile. The items with the highest similarity score are then recommended to the user. This method is quite effective, however a drawback is the inefficiency at classification time, since there is not a true training phase and therefore all computation is done at classification time. [27].

In the model-based approach, a model is created based on training data. For a new item, this model can predict if the user prefers it. A technique that has been widely used in content-based recommender systems is naïve Bayesian classification [31]. However

other methods are used as well. As Lops et al. [27] notices, empirically the naïve Bayesian classifier does a good job in classifying text documents, but nearest-neighbor classifiers or support vector machines can do a better job. In Pazzani and Billsus [32] a more extended overview on how the different techniques work can be found.

Problems

In both [2] and [32] some problems relating to content-based recommender systems are listed. One of the problems listed is the overspecialization problem. In content-based recommender systems it can happen that almost all similar items are recommended. Lets consider as an example a movie recommender system and a user that prefers the James Bond "Golden Eye" movie. For such a user, recommending five other James Bond movies might not be useful, as the user most likely will already know about the other movies and would therefore be able to find them herself. This can be improved by cross-domain recommendations, which will be described in section 2.3. Another way to improve it is proposed by Ziegler et al. [51], which introduces the Intra-List Similarity Metric to measure the similarity within a recommendation list to avoid similar items in a list.

Another problem, the limited content analysis problem is more a practical problem. It states that the analyzed content needs to contain enough information to discriminate items the user likes from the ones the user does not like. This can in general be solved by providing more information.

The final problem discussed here is the new user problem, which is a cold-start problem. The new user problem is the problem that a new user can not be provided with personalized recommendations until the system understands the user's preferences. This is discussed more in-depth in section 2.4.

2.2.2 Collaborative filtering recommender systems

According to [2], collaborative filtering recommender systems recommend the user items that people with similar tastes and preferences liked in the past. It is the most popular and used method. Early examples are the Tapestry system [15], GroupLens [35] [21], Ringo [43] and Bellcore's Video Recommender [18]. For collaborative filtering a user profile (see section 2.2.3) of the current user, other user profiles and a recommendation algorithm are needed. However, a property of the problem for which we propose a new solution is that there is no previous knowledge of users' preferences. Therefore standard collaborative filtering techniques are not useful to solve the problem. In the remainder of this section it will however be shortly introduced for completeness.

Recommendation algorithms

Collaborative filtering approaches are divided into heuristic-based and model-based approaches [46], as is also the case for content-based methods. Although, the difference is that in collaborative filtering the methods use ratings instead of content-based features to compute recommendations.

In heuristic-based methods, for collaborative filtering Pearson correlation is used as similarity measure quite often, next to the cosine-similarity measure. Several prob-

lems with heuristic-based methods and performance-improving modifications are listed in [2], [39] and [46].

For model-based approaches several algorithms from the field of machine learning have been applied to collaborative filtering. More information about the limitations, the performance of the different algorithms and the ability to address the challenges can be found in [39] and [46].

Instead of finding the most similar user, another approach is to compute similarities between items. This is called item-based collaborative filtering [39]. Sarwar et al. [38] has shown that item-based collaborative filtering is more accurate than user-based collaborative filtering.

Problems

The biggest problems related to collaborative filtering are cold-start problems and sparsity. Sparsity means that usually the number of known ratings is small compared to the number of ratings that need to be predicted. This problem is in general researched in the field of machine learning, leading to solutions like Singular Value Decomposition (SVD).

The cold-start problems are discussed more in depth in section 2.4. And information about other problems related to collaborative filtering methods can be found in [46].

2.2.3 User profiles

In both content-based and collaborative filtering recommender systems, a user profile is created to characterize the user for which recommendations are computed. This is in contrast to information retrieval systems where a user expresses her information need by a query.

In general a user profile contains information about the user's preferences. Therefore it either contains ratings for seen items or content-based features of items that are marked as relevant to the user. However, as Adomavicius and Tuzhilin [2] argues, most recommender systems do not take full advantage of other data about the user, like demographical information. This could for example be used to calculate neighbors in collaborative filtering.

The information stored in a user profile can be elicited explicitly or implicitly [32] [2]. A discussion about the pros and cons of explicit and implicit elicitation can be found in Schafer et al. [39].

2.2.4 Hybrid recommender systems

In the previous sections the two most used techniques in recommender systems, content-based and collaborative filtering, are discussed. Because both techniques have their drawbacks, lots of solutions have tried a combination of them to overcome some of the problems related to one of the techniques. These combinations are called hybrid recommender systems.

Adomavicius and Tuzhilin [2] have classified the approaches into four different categories:

1. Implement a content-based and collaborative filtering approach separately and combine the predictions;
2. Incorporate some content-based characteristics into a collaborative filtering approach;
3. Incorporate some collaborative characteristics into a content-based approach;
4. Construct a general unifying model that incorporates both content-based and collaborative characteristics.

For each of the categories they list some researches in that category. For example Fab [5] creates content-based user profiles to calculate similar users instead of calculating it based on rating profiles. This approach is also used in Degemmis et al. [12].

More information about hybrid recommender systems is contained in the survey of Burke [9].

2.3 Cross-domain recommender systems

The recommender systems discussed in section 2.2 all recommend items from a so-called single domain. However users' preferences may span across multiple domains and therefore dependencies between preferences in different domains can exist. This makes cross-domain recommender systems an interesting research field.

In the literature two cross-domain recommendation tasks are defined. The first task is to transfer and exploit user knowledge acquired in one domain into several other domains. This can help a user by recommending her with items in a novel, unexplored domain. This is possible because there might be dependencies between preferences in different domains. The second task is to recommend items from multiple domains in a joint recommendation. This can lead to more diverse recommended items. An example of a cross-domain recommender system that recommends items from multiple domains is Amazon, which recommends e.g. DVDs, music, books and video games. The first described task is in general seen as the true cross-domain recommendation task and is also the task researched in this thesis.

The first to research cross-domain recommendations were Winoto and Tang [49]. They verified the existence of correlations of users' preferences for items in different domains and argued that determining relations between different domains is the core of computing cross-domain recommendations. They also proposed a model to exploit users' preferences in a source domain to recommend items in a target domain. They showed that cross-domain recommendations are less accurate than single-domain recommendations, but cross-domain recommendations are more diverse, overcoming the overspecialization problem of content-based methods, and might therefore lead to a higher user satisfaction and engagement. Next to these advantages, cross-domain recommendations can address the cold-start problem and mitigate the sparsity problem.

However, it is not always that easy to compute cross-domain recommendations. There are different situations of relations between domains. For some situations it is harder to compute cross-domain recommendations than for others. Cremonesi et al. [10] defined the different situations of overlap (relations) between domains as follows:

1. **No overlap:** There is no overlap in items and users between domains;
2. **User overlap:** There are some users that have expressed their preferences in both domains. For example one user has expressed her preferences in a certain book and a certain movie;
3. **Item overlap:** There are some items preferred by users from both domains. For example two content providers share a catalog of items (e.g. two TV providers broadcasting the same set of TV channels);
4. **Full overlap:** There is overlap in items and users between domains.

Most research has been conducted on the overlap situations. Fernández-Tobías et al. [14] defines overlap relations between domains as "characteristics" that are shared by the user/item profiles in the different domains. These characteristics can be diverse, e.g. ratings or content features. Using these characteristics a content-based or collaborative filtering approach can be used to compute cross-domain recommendations. From these approaches, collaborative filtering is the most used technique for the overlapping situations. How collaborative filtering works in these situations is explained in the following example. Let us consider a cross-domain recommender system containing the domains books and movies and a user called Alice that prefers the book "Harry Potter". This user wants the system to recommend movies to her. A collaborative filtering technique will look for users that also like the book "Harry Potter". Let us consider that one of these users called Bob also prefers the movie "Lord of the Rings". Collaborative filtering assumes then that Alice who wants recommendations from the domain movies might also like the movie "Lord of the Rings" and recommends it to her. This example is visually supported by figure 2.2.

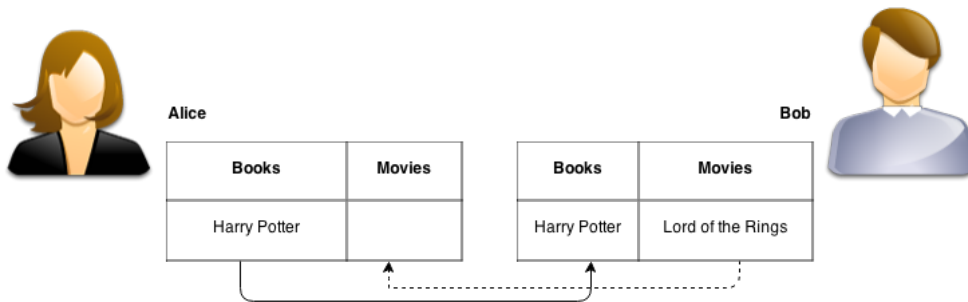


Figure 2.2: A cross-domain recommendation using collaborative filtering in a situation of overlap

However the no overlap situation can not be solved in such a straight forward way. As domains are often mutually exclusive (e.g. music and books), the no overlap situation should be solved using other approaches. Fernández-Tobías et al. [14] emphasizes that approaches have to be developed that find or build some type of explicit/implicit relations between domains, which would be used as semantic bridges connecting different domains in a recommender system. This is the situation researched in this thesis.

Several approaches have been proposed to compute cross-domain recommendations. Loizou [26] lists three of them:

1. Integrate and exploit explicit user preferences distributed in various systems;
2. Record user behavior and actions aiming to learn user preferences to use them for generating joint recommendations on multiple domains;
3. Combine recommendations from different domains to build a single system.

The first approach comes from the field of user modeling. The difficulty here is to cope with the different data formats, languages used or any other difference between systems. An example is the work of González et al. [16] who propose a domain-independent smart user model to relate profile characteristics of different domains. Another example is the work of Shapira et al. [42] who used user preferences from Facebook to generate an initial user profile, tackling the new user problem. Other methods proposed are to construct a unified user model using tags [1] or semantic modeling [48].

The second and third approach are not proposed that often. The goal of these approaches is to generate joint recommendations, which is out of the scope of this thesis.

Besides the approaches listed by Loizou [26], two newer approaches are listed by Fernández-Tobías et al. [14]:

1. Use social tags and semantic knowledge to establish relations between cross-domain user preferences and/or item attributes;
2. Apply transfer learning techniques investigated in machine learning to perform collaborative filtering where there is no explicit user/item overlap between domains.

The first approach uses several kind of content-based features to create relations between domains. Different approaches have been tried, such as extracting multi-domain knowledge from the Semantic Web (e.g. from Linked Data ontologies [13] [26]). But also social tags [19] [44] are used to create bridges between domains.

The second approach stems from the field of machine learning and tries to improve collaborative filtering in situations with no overlap between domains. An important work in this field is from Li et al. [25].

2.4 Cold-start problem

A cold-start problem arises when a new user, item or domain is added to a recommender system, creating a situation in which not enough information about users' preferences for items is available to come up with good recommendations. This situation can also occur when a complete new system is set up. In the literature different solutions have been proposed for the different situations. The new item problem is discussed in section 2.4.1. Next the new user problem is discussed in section 2.4.2, followed by the new domain problem 2.4.3. Finally the new system/new community problem is discussed in section 2.4.4.

2.4.1 New item

When a new item is added to a recommender system, no one has expressed her preference for that particular item yet, for example by rating the item. This is especially a problem in collaborative filtering recommender systems, since in that method an item needs to be rated by a substantial number of users before the recommender system will be able to recommend the item.

In content-based recommender systems this problem is less important, since new items are linked to other already existing items based on content using heuristic-based approaches or model-based approaches.

So, to solve the problem for collaborative filtering methods, often a hybrid solution is constructed so that new items are linked to other already rated items using the content-based methods.

However, it can happen that some new items can not be linked to already rated items using content-based methods and therefore might never be recommended. To overcome this, these items might be randomly recommended so that they are rated by some users after which it will be possible to recommend them to other users using collaborative filtering.

2.4.2 New user

When a new user enters a recommender system, not enough information about her preferences is known to come up with good recommendations. This is a problem that is relevant in both content-based recommender systems and collaborative filtering recommender systems. The new user problem is studied more often in literature than the new item problem.

Park [30] proposes a hybrid method exploiting not only user ratings but also features of items and users, like demographical data, to overcome cold-start problems. For collaborative filtering, when too less ratings are available, similar users are then computed based on demographical data.

A similar solution has been proposed by Lam et al. [23], where a vector aspect model with user information is used. The features used in the model are age, gender and job. The idea is based on stereotyping, so that people with the same features will also share the same preferences. Although it is a possible solution, it does not seem to be effective to a nearly one-million user data set. More critical attributes of users should be selected to improve the performance of the model.

Another approach, proposed by Sahebi and Cohen [37] use communities extracted from social networks to improve standard collaborative filtering technologies in a new user situation. They concentrate on the problem that a user is new in a certain system, but has a history in another system. The assumption they have made is that users within the same latent community are a better representative of similar user preferences in comparison with all users. From the gathered results it can be seen that it indeed slightly improves upon standard collaborative filtering.

Finally, Ahn [3] argues that the new user cold-start problem exists because of the heuristic similarity measures that do not deal well with less data and therefore proposes a new heuristic similarity measure, PIP (Proximity-Impact-Popularity). Compared to

traditional heuristic similarity measures, the proposed measure outperforms the traditional ones.

2.4.3 New domain

The new domain problem is only a problem in cross-domain recommender systems. It is a form of the new item problem, since actually it is nothing more than adding multiple new items to a system. However, these new added items are from another domain, making it harder to relate them to items already part of the recommender system, as different domains are expressed in different vocabularies. Therefore this situation might need another solution than the new item problem.

In literature this problem is not addressed yet. However Zhang et al. [50] underlines that for diverse items content-based methods are not always useful since the description of items is limited by the vocabulary used by the recommender system. Therefore they propose to solve this situation using social tagging. They propose an algorithm that keeps track of tags that the user prefers. New items (and thus also new domains) can easily be recommended in this way when it gets a tag, because using the tag it can be linked to other items and other domains.

2.4.4 New system/new community

The new system problem is a combination of the new user and new item problem. As Schafer et al. [39] notice, when no ratings are available in the beginning of a new system, i.e. no user preferences are known, other approaches than collaborative filtering should be used. For example, start with a set of ratings from another source outside the system to bootstrap the new system. However the problem is that organizations often keep that data private either for competitive advantage or privacy concerns.

2.5 Evaluating recommender systems

Evaluating recommender systems can be done in different ways. In the literature different methods are proposed: an offline experiment, a user study or an online experiment. Each of these methods has its advantages and disadvantages [41] which will be discussed in this section.

2.5.1 Offline experiments

The most used approach is an offline experiment. The reason for this is that the cost, both in time and money, for this kind of experiment is low. Therefore it is suitable to compare a lot of different algorithms. However, to perform an offline experiment test data is needed. For recommender systems, this means that a ground-truth is needed so that the results produced by the different recommendation algorithms can be compared to this ground-truth.

The most used metric in offline experiments is the accuracy of the algorithm. Both Shani and Gunawardana [41] and Herlocker et al. [17] give a rich overview of the different metrics to measure accuracy. The different metrics can be divided in measuring

the accuracy of rating's predictions, measuring the accuracy of usage predictions and measuring the accuracy of the ranking of items.

For measuring the accuracy of rating's predictions, Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) are the most used metrics. For the accuracy of usage predictions, precision, recall, F-measure and the Receiver Operating Characteristic (ROC) curve are used in general. Some alternatives to the standard ROC-curve are proposed by Schein et al. [40]. To measure the accuracy of the ranking, the (Normalized) Discounted Cumulative Gain (NDCG) is used quite often. Other used measures are Average Precision, the R-score proposed by Breese [8] and Mean Reciprocal Rank (MRR). To get to know more about the advantages and disadvantages of the different metrics the papers of Shani and Gunawardana [41] and Herlocker et al. [17] are a good follow-up.

After some years of measuring only accuracy, some researchers began to argue that accuracy is not telling the whole story, since it is not measuring the quality and usefulness of the recommendations. McNee et al. [29] show why accuracy metrics are not good enough and propose new user-centric directions for evaluating recommender systems. Shani and Gunawardana [41] go even further and list a whole set of properties that play a role when comparing different recommender systems. They not only list the properties, but also propose metrics for most of them. This is different than the work of Konstan and Riedl [22] that only describes the properties. However, for most of the properties listed, an offline experiment is unsuitable and should be measured using a questionnaire in a user study. This is one of the drawbacks of an offline experiment, namely that it is able to only answer a narrow set of questions.

Therefore an offline experiment should be used in the evaluation procedure to filter out inappropriate approaches, leaving a subset of algorithms for which a user study is conducted. It is costly to do a user study on a huge number of situations since a task must be repeated in a user study to make reliable conclusions. And therefore in a user study less situations can be tested. Related work to how user studies can be performed is described in the next subsection.

2.5.2 User studies

Although user studies can answer more questions than an offline experiment, the drawback is that it is expensive and/or time-consuming. An example of a user study that evaluates different recommender systems is done by Sinha and Swearingen [45]. Next to measuring the user satisfaction towards the different algorithms, they also measured the user-computer interaction.

They are not the only ones measuring user-computer interaction. Measuring the system performance is quite often done using a user study. This is in contradiction with measuring the algorithm performance, which is not often measured using a user study. One example of measuring the system performance is another research by Sinha and Swearingen [47]. Another example is the work of Knijnenburg et al. [20] who looked at evaluation from a behavioral point of view. They evaluate recommender systems on different aspects to measure the change in behavior of a user, while Pu et al. [33] [34] proposed a framework to evaluate the overall quality of a recommender system. This includes all kind of aspects also related to the interaction of the user with the recommender system. However in this thesis the focus is on the recommendations and

not on the system, and therefore these evaluation paradigms are out of the scope of this thesis.

How a good user study should be performed is described quite often in the literature. In this thesis we used the book from Robson [36].

2.5.3 Online experiments

An online experiment is the experiment that provides the strongest evidence of the value of a new algorithm because real users perform real tasks. This kind of experiment is used quite often in real life settings to compare a new algorithm to an alternative using A-B testing in which the traffic to the system is randomly distributed to the different alternatives.

This experiment is in general a follow up of an offline experiment or user study, because just testing a new algorithm online without knowing how good the recommendations are is too risky, knowing that bad recommendations might scare off users.

The drawback is that online experiments are time-consuming and enough traffic is needed to be able to come up with reliable conclusions.

Chapter 3

A Social Web based solution to bootstrap new domains

In the previous chapter the set of theoretical and technical tools related to the topics covered by this thesis are introduced. In section 3.1 we will discuss the existing tools that can solve the problem of bootstrapping new domains in cross-domain recommendations without previous knowledge about user's preferences. From this discussion it follows that the existing approaches have their limitations. Therefore we propose a new model which is discussed in section 3.2. Finally, the chapter is summarized in section 3.3.

3.1 Discussion of existing approaches

This thesis targets the problem of bootstrapping new domains in cross-domain recommendations without previous knowledge about user's preferences. Lets consider as an example a movie recommender system and a user that prefers the movie "Harry Potter". To this system, a new domain of books is added. Because the user's preferences towards books is unknown, it is hard to recommend the user books. It is important to notice that it is only known that the user prefers "Harry Potter". This is important because when there would have been a complete user profile containing for example information about the user's preference for science fiction movies, this could have been used to recommend her science fiction books.

Because a property of the problem is the lack of user profiles and knowledge about user's preferences, tools related to collaborative filtering techniques and user modeling are unsuitable to solve the problem. Therefore the existing approaches all construct relations between domains based on content-based features. Tools proposed in literature are to extract multi-domain semantic knowledge from the Semantic Web (section 3.1.1) or to use social tags (section 3.1.2) to create the relations between domains. Content-based methods used for single-domain recommendations based on syntactical features, have not been researched before. However, this does not seem to be useful as items from different domains will in general be described in a different vocabulary.

3.1.1 Semantic Web

"The Semantic Web is an extension of the current web in which information is given well-defined meaning, better enabling computers and people to work in cooperation." [6]. Part of the Semantic Web is the idea of Linked Data which describes a method of publishing structured data so that it can be interlinked and become more useful. One possibility for which it can be used is to link concepts from multiple domains. For example, DBpedia¹, which is a database containing structured Linked Data from Wikipedia, can be used to link music artists (domain: music) to places of interest (domain: architecture) [13].

Using these links it is possible to bootstrap new domains and compute cross-domain recommendations. The advantage of this method is that it contains structured data and is therefore easier to query.

The drawback of this method is that the cost, both in time and money, is quite high to create these repositories. Furthermore, to be able to create relations between items from different domains, these domains have to be represented as Linked Data belonging to mappable vocabularies. Finally, another drawback is that currently it most often only links concepts based on semantics, but does not understand dependencies in users' preferences. To explain this, let's consider as example a cross-domain recommender systems containing the domains movies and books. The domain movies contains a documentary about the football club Ajax, while the domain books contains a book about the football club Feyenoord. Using the Semantic Web, these items can be related to each other, because they are both about football clubs. However, since these football clubs are rivals, this would be a bad cross-domain recommendation for most users that prefer one of the two clubs.

Although Semantic Web solutions might be able to solve the problem (for some situations), they have some drawbacks and therefore in this thesis another solution is researched.

3.1.2 Social tags

Social tags are labels that can be assigned to all kind of items to describe them. For example they are used by Flickr to find photos and videos which have something in common. It is a quite often researched method to create content-based features to relate different domains. For example, a book can be tagged as science fiction and a movie can be tagged as science fiction, after which it is possible to create a cross-domain recommendation.

The advantage of tags is that they can describe different types of items, like images and videos. Another advantage is that in general they can be provided by a large pool of humans, so that the task of describing items is distributed.

One drawback is that it will take some time before the whole new domain is tagged as tags need to be provided by humans, making tags less suitable for bootstrapping new domains. Another drawback of tags is that in general only a few tags are assigned, not telling whole the story of an item, which might result in bad recommendations. Finally, when too much freedom is given in the assignment of tags, it can lead to a vocabulary mismatch. For example, one user can describe a video about airplanes with the tag

¹<http://wiki.dbpedia.org>

"plane", while another user can describe it with the tag "aircraft". Although this can be solved using a thesaurus, it is something to take care of or something to avoid by using a fixed vocabulary.

Besides not being suitable to bootstrap new domains, because they have to be provided by humans, tags seems to be a good solution to create relations between domains. Therefore this solution will be used as a baseline method in the case study described in chapter 4, 5 and 6.

3.2 Introducing our new model

To overcome some of the drawbacks of the existing approaches, we propose a new model to solve the problem of bootstrapping new domains in cross-domain recommender systems without previous knowledge of users' preferences. The new model uses an external source to gather missing users' preferences in a source system that are needed to relate a new domain to existing domains based on users' preferences. It uses the Social Web as external source. The Social Web is a set of social relations that link people through the World Wide Web [4] and some of the Social Web systems, e.g. YouTube, contain users' preferences. Such a Social Web system will be used as an auxiliary system to relate the new domain to the existing domains based on users' preferences. If in this auxiliary system there is knowledge about users' preferences in the new and existing domains, the model transfers this knowledge to the source system. Next, this knowledge is used in the source system to relate items from the new domain to items from the existing domains such that items from the new domain can be recommended to a user that prefers an item in one of the existing domains. The advantage is that this model uses already existing knowledge such that a new domain can immediately be bootstrapped. Therefore it is no solution to just upload items from the new domain to an auxiliary Social Web system such as YouTube as it will take time before users' preferences towards these items will be known.

The difficulty in this model lies in how to connect the source system to an auxiliary Social Web system so that the relations between domains can be transferred. This is not straight-forward as the set of items in the source system might be different from the set of items belonging to the auxiliary system. Therefore a mapping between the domains in the different systems has to be made.

The whole process can be formalized in the following way:

Item A and item B from the source system are related iff:

- Items C and D are items from the auxiliary system
- Item A is related to item C
- Item C is related to item D
- Item D is related to item B

The strength of the relation will be based on the strength of the relationships needed to construct the new one. So, the more similar that item A is to item C, the more similar that item B is to item D and the better the relation between item C and D, the better the constructed relation will be. To make a cross-domain recommendation, item A and item B need to be from different domains. The working of the model is visually explained in figure 3.1.

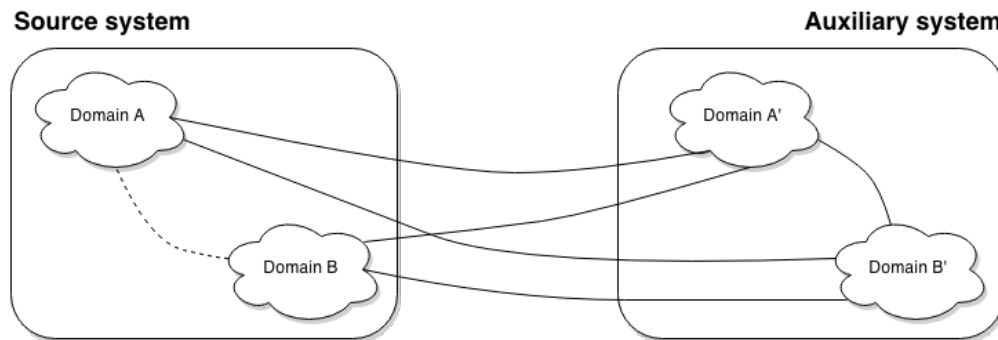


Figure 3.1: The new proposed model

Lets consider as an example a source system containing the domain movies and an auxiliary system containing the domains movies and books. The domain books is added to the source system. To create relations between this new added domain and the existing domain movies, the auxiliary system is used as it already contains relations between these domains. One of the books in the source system is the book "Harry Potter and the Prisoner of Azkaban" and one of the movies is "The Lord of the Rings: The Fellowship of the Ring". One of the books in the auxiliary system is the book "Harry Potter and the Goblet of Fire" and one of the movies is "The Lord of the Rings: The Two Towers". It is known in this auxiliary system that users that prefer the named book, also prefer the named movie creating a relation between these items. Although the items in the two systems are not the same, the model assumes that this same relation exists between the items in the source system, because the books and movies from both systems are quite similar.

An advantage of this solution is that the relations between domains are based on user's preferences and therefore only items from a new domain are recommended when there are dependencies between preferences in items from this new domain and existing domains. This is an advantage over the existing Semantic Web solution that does not take user's preferences into account. Another advantage of the solution is that it uses existing information and is therefore useful to bootstrap new domains. This information is (unconsciously) created by a lot of people. This is in contrast to Linked Open Data which has to be created consciously by less people.

Although the proposed model seems to be a good solution, it has some disadvantages. Most Social Web systems are closed systems, making it a black box for us. This makes it harder to understand why certain items are recommended. Furthermore the created relations are in general approximations as the source and auxiliary system do not contain the same items.

3.3 Summary

To summarize this chapter, a comparison between all possible solutions and the new proposed solution is given in table 3.1.

Model	Understands preferences	Data exists	Creation cost
Semantic Web model	-	+	-
Social Tags	-	-	-
Our model	+	+	+

Table 3.1: A comparison of the different solutions for the problem researched in this thesis

"Understands preferences" means that the model understands the dependencies between user's preferences. The Semantic Web model and social tags only relate items to each other based on content-based features, not taking into account what users prefer in multiple domains. However, our model takes this into account. "Data exists" means that the needed data exists at the moment that a new domain has to be bootstrapped. For the Semantic Web model and our model, this data is already there under the assumption that the domains needed are either represented as Linked Data belonging to mappable vocabularies or part of an auxiliary Social Web system. "Creation cost" means that the needed data has to be consciously created which is the case for the Semantic Web model and social tags, or is "unconsciously" created while users are using web systems.

The two disadvantages that come with our model are not named in the table as they are hard to compare to the other solutions. These disadvantages are that the auxiliary system is in general a black box and the created relations are an approximation of the relations in the auxiliary system.

Chapter 4

Case study: dataset

In the previous chapter, a new model to be researched has been proposed. This model is implemented in a case study to research its performance. In this chapter, the dataset used in the case study is discussed.

First, the requirements for that dataset are described in section 4.1. Next, the chosen dataset is described in section 4.2. In section 4.3 the taxonomy to split up the dataset in domains is described. To annotate the videos with a domain, an annotation task was set up. This annotation task and the results is described in section 4.4. Finally, the chapter is summarized in section 4.5.

4.1 Requirements

The chosen dataset must satisfy two requirements:

1. The dataset should contain items from multiple domains;
2. The dataset should contain tags for the items.

The reason for these requirements are that a cross-domain recommendation situation is tested and therefore the dataset should contain multiple domains. Furthermore as introduced in chapter 3, a tag-based solution is used as baseline. Therefore the dataset also needs to contain tags for the items.

4.2 Open Images dataset

The chosen dataset is the Open Images dataset. According to the website¹: "Open Images is an open media platform that offers online access to audiovisual archive material to stimulate creative reuse." All Open Images media items and the descriptions (metadata) can be accessed via an Open Archives Initiative Protocol for Metadata Harvesting (OAI-PMH) API². The metadata can be retrieved in two different formats. The

¹<http://openbeelden.nl/>

²<http://openbeelden.nl/api/>

Dublin Core format³ or the Open Images format, which is a mixture of DC Terms⁴ and an XML interpretation of ccREL⁵.

On the 25th of September 2013, we retrieved the dataset from Open Images using the available APIs. The Open Images metadata format was chosen, because it contains more detailed information compared to the Dublin Core format. For a full overview of all elements in the Open Images metadata format see appendix B. In total, the metadata of 2544 videos were retrieved.

Next, we retrieved the tags belonging to these videos. In total 499 unique tags assigned to at least one video were retrieved. Luckily, almost all videos have at least one tag. Only 20 videos do not have a tag.

4.3 Domain taxonomy for the Open Images dataset

There are different ways how the dataset could be split up in domains. We came up with the ideas of splitting up the dataset based on video types or categories, as those seems to be the most suitable for the Open Images dataset, and we created a general taxonomy for both ideas, see table 4.1 and table 4.2. Since the taxonomy based on categories is a taxonomy that is used quite often we decided that definitions are not needed in that case. Lets consider as an example a recommender system that uses the taxonomy based on video types and a user that prefers the documentary "Bowling for Columbine". We can recommend this user a news item about someone being shot in the United States. Lets consider as another example a recommender system that uses the taxonomy based on categories and a user that prefers a music video from The Beatles (domain:music). We can recommend that user a video about John Lennon from the domain people.

From the two proposed taxonomies, only one could be used in this case study to split up the dataset in domains. Therefore we had to decide which of the two would be better. Since the dataset does not contain either the type of the videos (according to our definition of type) nor the category (see appendix B) as metadata, manual annotation was required.

From our own experience with the videos in the Open Images dataset, the hypothesis arose that it would be easier to annotate the "type" of a video, than to annotate its category. It is important to annotate an item with only one type or category as an item can only belong to one domain according to the definition. The hypothesis arose as quite often it seemed that a video could be annotated with multiple categories. For example a video about driverless cars can be annotated with both the category technology and autos & vehicles. To test this hypothesis a small manual annotation task was set up.

In total, 20 videos were randomly selected to be annotated by 3 different users, so in total 60 data points were collected. The participants were told to annotate as many videos as they wanted. In the end, 7 participants were required to annotate all videos.

The participants were asked to annotate the video with one or multiple video types and one or multiple categories. To see if the taxonomies were complete, the option of

³<http://dublincore.org/documents/dces/>

⁴<http://dublincore.org/documents/dcmi-terms/>

⁵<http://www.w3.org/Submission/ccREL/>

Table 4.1: Taxonomy of types of videos

Type	Definition
Documentary/report	In a documentary or report, a real life phenomenon or item is viewed from different angles. Examples are: Bowling for Columbine and Fahrenheit 9/11.
Event coverage	An event coverage shows a real life event. There might be comments about what you see. Examples are: Ajax - Feyenoord and Lowlands.
Films	A film is a visual display of a story line made by a film producer. Examples are: The Godfather and Star Wars.
News item	In a news item a story is told about something that happened that day or week. Examples are: "Search area lost plane MH370 is narrowed" and "New international debate about Ukraine"
Series	Series are regularly returning shows. This also includes talk shows and game shows. Examples are: Game of Thrones and Jeopardy!
Video blog	In a video blog, someone can post diary entries about (their personal) experiences, observations or hobbies. Examples are: blogger Nigahiga on YouTube and Fred's Channel on YouTube.

providing another type or category was offered as well. Majority voting was used to compute the best video type and category for a certain video. On average 1.18 video types (sd: 0.42) were annotated per participant per video, and 1.78 categories (sd: 0.86) were annotated. This confirms our hypothesis that it would be easier to annotate a domain based on the type of a video than to annotate a domain based on its category, as a video can only belong to one domain.

Next, the distribution into video types and categories was calculated using majority voting. In table 4.3 an overview of the final results for the video types can be found and in table 4.4 an overview of the final results for the categories can be found.

As can be seen in the tables, for 4 videos no video type could be assigned, while for 7 videos no category could be assigned. There are two reasons why no video type or category could be assigned. First, none of the video types/categories got a majority vote, or second, multiple video types/categories got a majority vote and the video could therefore not be assigned one video type or category and is useless according to the definition that an item can only belong to one domain.

The reason that 4 videos could not be assigned a video type is that none of the video types got a majority vote. The reason that 7 videos could not be assigned a category is in 4 cases that none of the categories got a majority vote and in 3 cases multiple categories got a majority vote with the majority being equally large. Again this is in line with the hypothesis that it is easier to annotate one video type to a video than to annotate one category to a video.

So, to conclude, using majority voting for video types resulted more often in a result than for categories. Together with the fact that less video types than categories

Table 4.2: Taxonomy of categories of videos

Category
Animals & Pets
Art
Autos & Vehicles
Comedy & Entertainment
Culture
Education
Gaming
How to & Style
Music
People
Politics
Science & Technology
Sports
Traveling

Table 4.3: Video type annotations

Type	Number of occurrences
Documentary/report	6
Event coverage	3
News item	4
Video blog	3
Total	16

Table 4.4: Category annotations

Category	Number of occurrences
Education	2
Animals & Pets	2
Culture	4
Sport	1
Science & Technology	4
Total	13

are assigned per microtask and that an item can only belong to one domain according to the definition, the taxonomy used is the one based on video types. Therefore the definition of domain used in this case study is:

Definition. *A domain is the name of a group of items characterized by its video type. The domains are determined initially and items are matched if they fit in a domain. An item can only belong to one domain.*

Using this definition, an example of a documentary/report in the dataset is the video "Collectie spaarpotten in Amsterdam", which is a documentary/report about a

certain collection of piggy banks. An example of a news item is "Branden vernielen honderden hectare natuurgebied", which is a news item about forest fires.

A query on the metadata of both videos shows that both contain the description: "Bioscoopjournaals waarin Nederlandse onderwerpen van een bepaalde week worden gepresenteerd" (English translation: Newsreels in which Dutch subjects of a certain week are presented). This is a feature of the dataset, that it is all archive material from old newsreels. However, in the old days, the newsreels did not only contain news items according to our definition. Using automatic annotation methods, both would be classified as news item, which is wrong.

A query on our dataset tells us that this is the case for 2009 of the 2544 videos. Therefore automatic annotation methods are unsuitable for our case study and we decided to set up a manual annotation task. We use the taxonomy proposed in table 4.1 for the manual annotation task as none of the participants in the experiment filled in an additional video type.

4.4 Domain annotation task

The annotation task has two goals:

1. Assign a domain to every video;
2. Measure the level of agreement about a domain.

The second goal stems from the observation, during the exploration phase of the Open Images dataset and the experiment performed to decide on the taxonomy, that for some videos the domain is more clear than for other videos. This level of agreement might later be used as input for the model.

With these goals in mind we set up the annotation task. For every video we asked multiple participants to annotate it with one or multiple domains. The reason that we asked multiple participants is to increase the reliability. We used majority voting to calculate the best domain for a video. This also gave the level of agreement, because it could be calculated how many of the asked participants agreed on a certain domain.

We decided to measure the level of agreement in this way, because it is easier for a participant to annotate multiple domains when she is unsure, instead of letting the participant annotate a video with one domain and ask her to express the certainty in a number.

Next to choosing one of the domains from the taxonomy, the participants were also given the option to annotate another domain to a video via a textbox, like we did in the previous experiment to decide on the taxonomy. This was done to see if a domain was missing in the taxonomy and should be added.

This annotation task was created using the platform BruteForce. This platform is a crowd sourcing platform that allows to utilize high quality labor and is created by a graduate student of Delft University of Technology. The advantage of using this platform over existing crowd sourcing platforms like Mechanical Turk or CrowdFlower is that it allows you to distribute the task yourself. Therefore it is possible to choose the crowd yourself. Using acquaintances to solve the tasks makes it cheaper than paying workers on Mechanical Turk or CrowdFlower. Another advantage of distributing

the task under acquaintances, instead of anonymous workers, is that there will be less spammers as acquaintances are more trusted.

The task to annotate one video is called a microtask. In each microtask, the following instruction text was given to the participant, see also figure 4.1:

In this experiment you will have to annotate a series of videos. Each video is different qua length, quality and some will include sound, others will not. You can decide how much time you take for the experiment since you can stop after each video. If you want to continue at a later point in time, this is no problem.

For each video, you will be asked to choose the type and category where you think the video belongs to. Select the option(s) of your choice. If your choice is not in the list, use the 'other' field. If you do not fully understand one of the types, read the explanation below it.

Note: if a video contains parts of a film, it is still a film.


As can be seen in the instruction text, participants were allowed to decide themselves how many microtasks they wanted to solve. It can also be seen that the word 'type' is used instead of 'domain' as this is assumed to be more clear for the participants. And finally, although we decided to use video type as domains, the participants were asked to annotate one or multiple categories as backup so that we could switch the taxonomy in the case it turned out that video type was not fitting.

To decide how many videos could be annotated, we used the time taken to solve a microtask in the experiment to decide on the taxonomy as this task was the same as the just described manual annotation task. The average time taken to solve a microtask was 87.27 seconds (sd: 85.73 seconds). Assuming that participants are willing to spend 20 minutes of their spare time to help, as this is a quite often used number, they will solve 13.75 microtasks on average. To solve the whole dataset of 2544 videos, in that case 555 participants are needed. Using only acquaintances, this number of participants seems to be pretty unreachable. Therefore it was decided not to annotate the complete dataset. It was estimated that about 100-125 participants could be recruited. Using the estimates of 20 minutes per participant and 13.75 microtasks, we decided to set up the annotation task for 500 videos (1500 microtasks).

So, we randomly selected 500 videos and set up the task. We distributed the task using e-mail, Facebook, Twitter and word-of-mouth advertising. Everybody was suitable to solve a microtask. The only constraint was to understand Dutch, because the language spoken in most of the videos is Dutch.

4.4.1 Results annotation task

The task ran for 21 days, after which 356 videos were annotated by one participant, 129 videos were annotated by two participants, 13 videos were annotated by three participants and 2 videos were not annotated. This distribution arose because of the way BruteForce handles multiple participants for one video.



Documentary/report
In a documentary or report, a real life phenomenon or item is viewed from different angles. Examples are: Bowling for Columbine and Fahrenheit 9/11.

Event coverage
An event coverage shows a real life event. There might be comments about what you see. The difference with news is that events might also be shown in a news item, but then it is a summary of the event and commented by a news reporter. Examples are: Ajax – Feyenoord and Lowlands.

Films
A film is a visual display of a story line made by a film producer. Examples are: The Godfather and Star Wars.

News item
In a news item a story is told about something that happened that day or week. Examples are: "Search area lost plane MH370 is narrowed" and "New international debate about Ukraine"

Series
Series are regularly returning shows. This also includes talk shows and game shows. Examples are: Game of Thrones and Jeopardy!

Video blog
In a video blog, someone can post diary entries about (their personal) experiences, observations or hobbies. Examples are: blogger Nigahiga on YouTube and Fred's Channel on YouTube.

Other:

Select the category (You can choose multiple categories if required)

Animal and Pets
 Art
 Autos & Vehicles
 Comedy & Entertainment
 Culture
 Education
 Gaming
 How to & Style
 Music
 People
 Politics
 Science & Technology
 Sports
 Travelling

Other:

Figure 4.1: An overview of the annotation task

So, in total 653 microtasks were done. It took 88 sessions to solve these microtasks, so on average 7.42 (sd: 12.86) microtasks were solved per session. Assuming that almost all participants started one session (as this is not measured by BruteForce), this is lower than expected. There can be two reasons for this. One, people spent less than 20 minutes of their time or two, the mean time spent per microtask is higher than in the experiment to decide on the taxonomy. After filtering outliers, the mean time spent per microtask is 85.13 seconds (sd: 57.95 seconds) which is in line with the results from the earlier experiment. Therefore we conclude that people spent less than 20 minutes of their time. This is in line with what we heard from the participants as they perceived the task as being boring.

Because the task was not completed, we took a closer look to decide if the results could be used or more annotations should be gathered. For the 129 videos annotated by two participants, there was agreement for 58 videos (44.96%). For 63 videos, both participants annotated one domain, but disagreed and for 8 videos multiple domains were assigned by one or both participants, however none of them were annotated by the other participant. For the 13 videos annotated by three participants, all three par-

ticipants agreed on the domain for 6 videos (46.15%). In another 5 videos, two participants agreed on a domain, so that in 11 of the 13 videos (84.62%) a domain could be assigned by majority voting.

Based on these results we decided that more annotations would be needed to create a reliable dataset with more videos. Therefore the 63 videos annotated by two participants that disagreed were assessed by another annotator. For 43 videos, this third participant agreed with one of the two annotators, creating a majority. Furthermore from the 356 videos with one annotator, 102 videos were annotated by two more participants. In 81 of these videos a majority agreed on one domain. The reason that so many videos were annotated by only a couple of participants is that these participants were experts and could therefore annotate the videos faster.

Aggregating everything together resulted in 254 videos being annotated by one participant, 66 videos being annotated by two participants, 178 videos being annotated by three participants and 2 videos were not being annotated at all. From the 66 videos that were annotated by two participants, 58 videos could be assigned to a single domain using majority voting. From the 178 videos that are annotated by three participants, 135 videos could be assigned to a single domain using majority voting. So the total dataset that we created contains 193 videos. The distribution of these videos in the different domains can be seen in table 4.5.

Table 4.5: Final distribution of domains

Domain	Number of videos
Documentary/report	75
Event coverage	55
News item	60
Films	0
Series	0
Video blog	3
Total	193

The second goal of the annotation task was to measure the level of agreement, see figure 4.2.

For the 135 videos that were annotated by three participants and could be assigned to a single domain using majority voting, for 27 videos all three participants agreed on a domain. In 108 videos, two participants agreed on a domain. For the videos that are annotated by two participants and could be assigned a domain using majority voting, the level of agreement can not be computed as the third participant can agree or disagree with the other two participants. Therefore we decided not to incorporate the level of agreement in the model anymore.

Finally, participants could also annotate the videos with other domains. In total, in 5 microtasks the option to name another domain was used. Since there was no overlap between these other domains they were not added to the taxonomy.

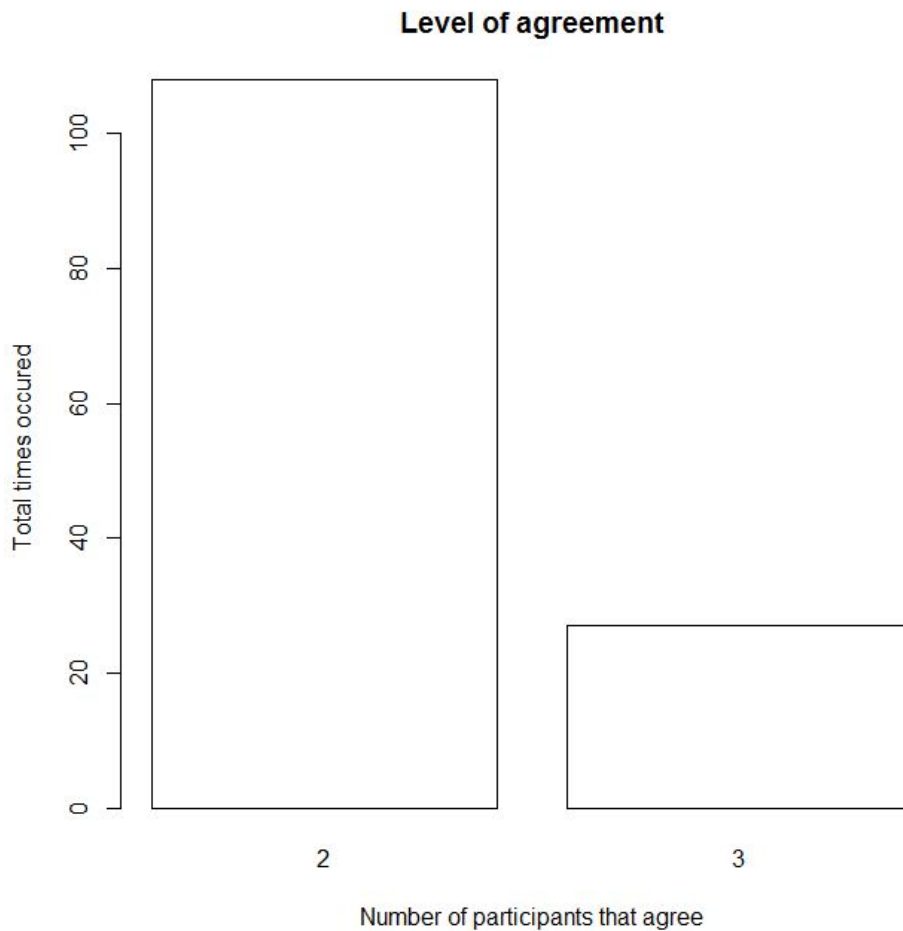


Figure 4.2: The level of agreement

4.5 Summary

To summarize this section, we decided to use the Open Images dataset in the case study because this dataset of archive videos also contains tags for the videos. The disadvantage of this dataset is that it needed manual annotation for domains. Based on a small experiment it turned out that splitting up the dataset based on video types would be the most useful way to identify domains. As automatic annotation methods would cause problems on this dataset, we decided to set up a manual annotation task. In total 193 videos were annotated with one single domain. The distribution of the videos in the domains is given in table 4.5. These videos are used in the rest of the case study.

Chapter 5

Case study: implementation

In this chapter, the implementation of the new model proposed in chapter 3 is discussed. First the general implementation is described in section 5.1. In this implementation several parameters needs to be tuned. To tune these parameters an offline experiment is set up. This set up is discussed in section 5.2. Finally, the results of this experiment are discussed in section 5.3.

5.1 Implementation of the new introduced model

The implementation of the model is built up in different phases. First an auxiliary system had to be chosen from the Social Web that contains the relations that needs to be transferred. This auxiliary system is discussed in section 5.1.1. Because the auxiliary system does not exactly contain the same items as the Open Images dataset, a mapping had to be made to find the most similar item in the auxiliary system for each video in the Open Images dataset. This mapping is discussed in section 5.1.2. Next, the relations in the auxiliary system had to be retrieved. This is discussed in section 5.1.3. These related items in the auxiliary system had to be mapped to the most similar items in the Open Images dataset. This mapping is discussed in section 5.1.4. These different parts are visually displayed in figure 5.1. Finally, the strength of the created relation between two Open Images videos had to be calculated. The calculation used is discussed in section 5.1.5.

5.1.1 Auxiliary system

The chosen auxiliary system should fulfill the following requirements:

1. The auxiliary system contains at least the same domains as the Open Images dataset;
2. The auxiliary system contains relations between these domains;
3. The relations between the domains in the auxiliary system should be freely available.

We chose the biggest video website in the world, YouTube, as our auxiliary system. YouTube contains videos from the domains news items, documentaries/reports and

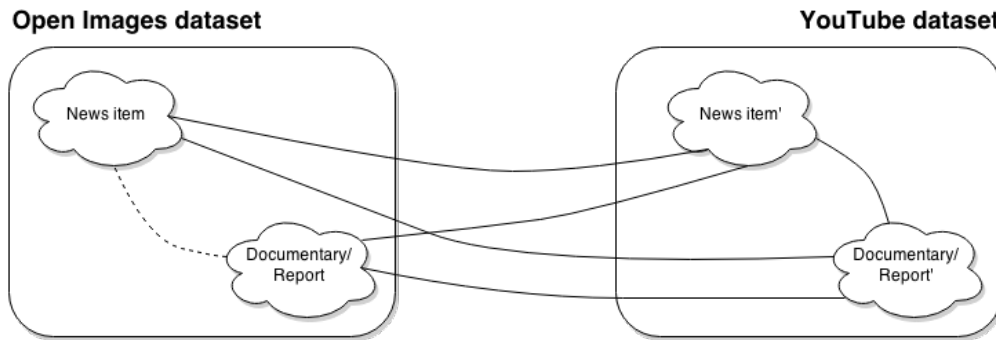


Figure 5.1: Our model applied on the Open Images dataset with the chosen auxiliary system using the domains news item and documentary/report as example

event coverages, the only domains with enough videos left in our dataset. We assumed that it would also contain the relations that are needed. Finally, YouTube provides an API¹ to access its database and retrieve the needed relations if they are present.

5.1.2 Mapping from Open Images to YouTube

Because YouTube does not contain the same items as the Open Images dataset, a mapping had to be made from the Open Images videos to the most similar YouTube videos. The only way to find videos on YouTube is to query its database. We used the YouTube API for this purpose. Since there is no literature on the working of the search system of YouTube, we assumed that YouTube tries to find the items that best fits a given input query. Given this assumption, the 50 most similar videos can be retrieved for a given input string.

Based on the metadata of the Open Images videos (see appendix B) several inputs can be generated. The most logical options seems to be to use title, title and date, or title and domain of an Open Image video as input. For the 193 videos in our dataset we queried YouTube using these different inputs. We measured how often an input did give no search results and the average number of search results. These statistics are displayed in table 5.1.

Input query	# no result	Avg. # of results
Title	58 (30,1%)	32.57 (sd: 20.24)
Title & date	109 (56,5%)	11.62 (sd: 12.74)
Title & domain	98 (50,5%)	19.86 (sd: 19.92)

Table 5.1: Statistics for different input queries on YouTube

Even though the strings are given as input without the strict option (using quotes: "some string"), quite often YouTube does not find a result. Given the closed nature of the YouTube system, it is not easy to justify this result. To get a gut feeling about the working of the YouTube search system, we set up a small experiment. A summary of

¹<https://developers.google.com/youtube/v3/>

the results of this experiment is reported in appendix D. It seems that the search system is not working similar to most information retrieval systems. For example, using the Dutch query "boek" or "een boek" already gives totally different results, where in most information retrieval systems the stop words are filtered out giving the same results for these queries. Another example is using the query "the boy's car" or "the boy's cars". This also gives totally different results, where in most information retrieval systems queries are stemmed, this does not seem to be the case in the YouTube system.

Finally, we want to know the similarity between the input Open Images video and the retrieved YouTube videos such that it can be used to calculate the final strength of the relation that needs to be created. However, this information can not be retrieved from YouTube. It could however be calculated by ourselves, for example using cosine similarity, but we decided to investigate this as part of the future work.

5.1.3 Retrieving the relations from YouTube

After the most similar YouTube videos are retrieved for a certain Open Images video, the relation between the retrieved videos and other YouTube videos needs to be retrieved. The related videos can be retrieved for a given YouTube video using the YouTube API. For a given YouTube video, 50 related videos can be retrieved.

According to [11], videos are related if they co-occurred in a session in the last 24 hours and otherwise related videos are calculated content-based. However, this publication is from 2010 and might be outdated already. Since then, no other information about the definition of relationship used by YouTube is published.

5.1.4 Mapping from YouTube to Open Images

The final step in the implementation of the model is to map the related videos back to Open Images videos. As the videos are not exactly the same as the Open Images videos, the most similar Open Images have to be calculated.

To compute this similarity, the metadata of the YouTube videos (appendix C) has to be compared to the metadata of the Open Images videos (appendix B). We decided to compare the videos based on title and description. This has been decided because the metadata of most retrieved YouTube videos do not contain the recording date. In general only the publication date is set. Because YouTube is a dynamical system, the metadata of the related videos is retrieved once using the title, title & date and title & domain of the Open Images videos as input.

Because the metadata to compare contains unstructured text, we constructed TF-IDF vectors [28] to compare YouTube videos with Open Images videos. We used Lucene to construct these TF-IDF vectors. See figure 5.2 for the pipeline used by Lucene to create these vectors.



Figure 5.2: The Lucene pipeline to create TF-IDF vectors from an unstructured text

Using these vectors, we used nearest-neighbor classification to compute the most similar Open Images video for a given YouTube video, as it followed from literature that nearest-neighbor classifiers do a better job than naive Bayes classifiers (model-based technique) and that the only drawback is that all computation is done at classification time making it slightly inefficient. However since we are dealing with a relatively small dataset this is not a big issue.

Again Lucene is used to compute the most similar item(s) for a given YouTube video using nearest-neighbor classification. Lucene uses a slightly changed version of cosine-similarity for the scoring². As we wanted to find the item(s) that is most similar on title and description, a faceted search is done, using the title and description of the YouTube video as input query.

5.1.5 Scoring function

The strength of a relation between two Open Images videos, a source and target video, from two different domains is based on the similarity between the source video and the found YouTube videos, the strength of the retrieved relation from YouTube and the similarity between the related YouTube videos and the target video on Open Images.

However, the similarity between a video from Open Images and its search results on YouTube can not be gathered using the YouTube API. The same holds for the strength of the relation between two YouTube videos. Therefore only the position of a YouTube video in the search results and the position of a YouTube video in the related videos ranking will be taken into account, leading to the following scoring function, with "similarity" being the similarity score computed by Lucene:

$$\text{score} = \text{similarity} * \frac{1}{\text{position in search result}} * \frac{1}{\text{position in related video ranking}}$$

This scoring function might punish search results or related videos with a lower ranking to hard. Therefore we created an alternative smoothed version of this scoring function. In this alternative, the logarithm of the position in the search result and the position in the related video ranking is used. As this will give problems for the first search result and first related video, every position is added by one. This leads to the following function, with "similarity" being the similarity score computed by Lucene:

$$\text{score} = \text{similarity} * \frac{1}{\log(\text{position in search result} + 1)} * \frac{1}{\log(\text{position in related video ranking} + 1)}$$

It is no problem to just add one, because we do not care about the real value of the score. The score is only used to order the videos and the videos with the highest scores are recommended.

²https://lucene.apache.org/core/4_0_0/core/org/apache/lucene/search/similarities/TFIDFSimilarity.html

5.2 Tuning the model parameters in an offline experiment

In this section we discuss the set up of the offline experiment that we performed to tune the different parameters in the implementation of the model. According to [41], it is quite common to use an offline experiment to tune the parameters of a model, after which the best tuned models continue to the next phase of evaluation. The reason is that offline experiments are less expensive than for example user studies.

The outcome of this offline experiment is used as input for the user study described in the next chapter. We decided to compare different implementations of our model, that we will call strategies, in the user study to evaluate the performance of our model. The different strategies that we will compare are based on the different inputs to search on YouTube, namely title, title & date and title & domain.

The parameters for each strategy are tuned on different videos. From each domain from the Open Images dataset that contains enough videos (news item, event coverage and documentary/report) we randomly selected 5 videos. So in total, 15 different videos are used to tune the parameters. For each of these videos we computed the 5 best cross-domain recommendations. For the videos from the domains news item and event coverage we created a cross-domain recommendation to the domain of documentary/report, while for the videos from the domain documentary/report we created a cross-domain recommendation to the domain of news item. So, for each source domain, the target domain is chosen that contains the most videos. This is done to increase the change of having relevant items as the dataset is already quite small.

The parameters that are tuned in this offline experiment are described in section 5.2.1. Next, to compare the different values for a parameter we need a ground truth to be able to say something about the accuracy of the recommendations created using the different values. This ground truth is described in section 5.2.2. Finally, the metric used to compare the recommendations with the ground truth is discussed in section 5.2.3.

5.2.1 Parameters

The parameters that can be tuned in the model can take several values. Since the number of values each parameters can take is enormous, we decided to only compare the performance of a couple of values for each parameter. The values compared are given in table 5.2.

Parameter	Value 1	Value 2	Value 3
Number of search results	50	20	1
Number of related videos	50	5	1
Number of similar Open Images videos given a YouTube video	5	1	-
Scoring function	No-log-smoothing	Log-smoothing	-

Table 5.2: Values to compare for each parameter

For the number of search results we decided to compare 1 (the minimum) with 20 (the standard number shown on YouTube) and 50 (the maximum). For the number of

related videos we decided to compare 1 (the minimum) with 5 (the number of recommendations we want to give as this is quite standard) and 50 (the maximum) . For the number of similar Open Images videos for a given YouTube video we compare 5 (the number of recommendations we want to give) with 1 (the minimum). Finally, we compare the different scoring functions proposed in the previous section.

For each strategy these parameters are tuned. The number of combinations of these parameters is 36, leading to 108 combinations to test. As this is quite time consuming we decided to constantly change one parameter, assuming the parameters are independent. This resulted in testing 10 combinations for each strategy.

5.2.2 Ground truth

Using each value of the different parameters cross-domain recommendations are calculated for 5 randomly selected videos for each of the domains news item, event coverage and documentary/report, so in total it is compared on 15 videos. To be able to say something about the accuracy of the different recommendations we need a ground truth that says for each of these 15 videos which of the possible recommendations is relevant and which is not. For the videos from the domains news item and event coverage we created recommendations from the domain documentary/report. This domain contains 75 items to recommend. For the videos from the domain documentary/report we recommend items from the domain news item. This domain contains 60 items to recommend. So in total we needed a ground truth that contains relevant judgments for 1050 pairs.

For each pair containing a source video and a possible recommendation we want a judgment that says if the recommendation is relevant for the given source video. Unfortunately no data is collected by the Open Images platform that can be used for this relevance judgment. The only collected data that tells something about videos that are watched after each other and might therefore be judged as relevant is the Google Analytics data of the Open Images platform. However, this platform already implemented tag-based recommendations and therefore the Google Analytics data can not be used as it is biased towards tags.

Therefore we tried to create the judgments by ourselves. To create an unambiguous relevance judgment, for each pair we use the following definition of relevant:

Definition. *A recommendation is relevant for a video X if the recommended item shares one of the following features with video X: city, both are silent videos, or they share at least one category.*

Although relevant is different for every person, we tried to generalize it as much as possible to make it apply to most people. As we want to create recommendations that are most satisfactory to most people, we made a list of possible information needs. These information needs are:

- Someone wants to find all videos about a certain topic;
- Someone wants to find all videos about a certain city;
- Someone wants to find all silent videos.

For example, someone wants to find all videos about soccer. Another example is that someone is interested in the history of the city of Amsterdam and therefore wants to find the videos from all domains about Amsterdam. Or someone wants to reuse all videos without sound and wants to find all those videos. Although it might not be a correct definition of relevancy for other datasets, we think these information needs occur most for the Open Images dataset.

For example, lets consider a user that prefers the video "Hardloop wedstrijd in het Vondelpark" from the domain event coverage. For this user a relevant cross-domain recommendation is the video "Vondelpark" from the domain documentary/report.

So, for each of the 15 videos, we created a ground truth. The number of relevant items for each of these videos is given in table 5.3.

Video	#items in target domain	#relevant items
News item 1	75	0
News item 2	75	2
News item 3	75	0
News item 4	75	21
News item 5	75	3
Documentary/report 1	60	3
Documentary/report 2	60	2
Documentary/report 3	60	6
Documentary/report 4	60	0
Documentary/report 5	60	1
Event coverage 1	75	4
Event coverage 2	75	2
Event coverage 3	75	21
Event coverage 4	75	23
Event coverage 5	75	1

Table 5.3: Number of relevant items for each video

The accuracy of the recommendations for each of these videos is measured. The videos for which there are no relevant items are filtered out, leaving 12 videos to measure the accuracy of the different implementations of the model.

5.2.3 Metric

We want to measure the accuracy of the recommendations that are created using the different values for each parameter. These accuracies are then compared to decide which value of the parameter gives the highest accuracy.

As we want a top-5 recommendation list that contains as many relevant items as possible with the relevant items on top, we want to measure the accuracy of usage predictions and the accuracy of the ranking. However, since for a lot of videos we have only a couple of relevant items (see table 5.3) measuring the ranking will be more useful.

As we have only relevant/non-relevant judgments, Average Precision and Mean Reciprocal Rank (MRR) are good metrics to use. However, there can be multiple

relevant items in the recommended list. MRR does not take this into account and therefore we use Average Precision as metric.

5.3 Discussion of the results of the offline experiment

In this section the results of the offline experiment are discussed. First, the parameters of the strategy that we will call Title are tuned. This is discussed in section 5.3.1. Next, the parameters of the strategy Title & Date are tuned in section 5.3.2. Finally the parameters of the strategy Title & Domain are tuned in section 5.3.3.

5.3.1 Strategy Title

We started to tune the strategy Title with tuning the number of search results we get back from YouTube. We measure the Average Precision for retrieving 50, 20 and 1 search result. The other parameters are kept constant. The Average Precision is measured on 12 videos. The results for each of these videos is presented in appendix E.

Over these 12 videos the Mean Average Precision (MAP) is taken. It can be seen from table 5.4 that retrieving 1 search result gives the best performance. This can be explained by the assumption that YouTube lists the most similar item on top. Adding more search results will then add more noise giving worse results.

Next, we tuned the number of related videos that we can retrieve on YouTube. The MAP-scores are given for 50, 5 and 1 related video in table 5.4. Clearly taking more related videos into account gives a better performance. We assume this is the case because taking more related videos into account increases the chance of finding a video that is also in the Open Images dataset.

Next, we tuned the number of similar Open Images videos that we want to find for a given YouTube video. We retrieved the MAP-scores for 5 and 1 similar videos. It follows from the results that only retrieving 1 similar Open Images video gives a better result. This might be explained by the assumption that retrieving more similar videos will add noise.

Finally, we tested the different scoring functions. The results tells us that log-smoothing the scoring function gives better results. As we retrieved only one search result and 50 related videos in this scenario, this result might be explained by the assumption that lower ranked videos in the related videos list are not necessarily worse videos. Punishing these lower ranked videos harder by using no-log-smoothing function clearly gives worse results.

So, to summarize, for the strategy Title we retrieve only 1 search result from YouTube. For this single search result, we retrieve its 50 related videos. For these 50 related videos we find the most similar item in the Open Images dataset and for the final scoring function we use the log-smoothing function.

5.3.2 Strategy Title & Date

Next, we tuned the parameters of the strategy Title & Date. Again, first we tuned the parameter search results. Also for this strategy it is best to retrieved only 1 search result, see table 5.5. The same holds for the related videos, again it is best to retrieve 50

Parameter	Value 1	Value 2	Value 3
Number of search results (50/20/1)	0.05	0.07	0.08
Number of related videos (50/5/1)	0.08	0.03	0.03
Number of similar Open Images videos given a YouTube video (5/1)	0.02	0.08	-
Scoring function (no-smoothing/smoothing)	0.08	0.13	-

Table 5.4: Mean Average Precision (MAP) for the different values of the different parameters

related videos. We can also best only find one most similar item in the Open Images dataset for a given YouTube video. Only for the scoring function the MAP-scores were closer for this strategy. However this is because of a rounding error. The MAP-scores are 0.1660 for the no-log-smoothing function and 0.1749 for the log-smoothing function. So, again, the smoothing function is better to use.

Parameter	Value 1	Value 2	Value 3
Number of search results (50/20/1)	0.04	0.04	0.17
Number of related videos (50/5/1)	0.17	0.11	0.12
Number of similar Open Images videos given a YouTube video (5/1)	0.02	0.17	-
Scoring function (no-smoothing/smoothing)	0.17	0.17	-

Table 5.5: Mean Average Precision (MAP) for the different values of the different parameters

So, to summarize, for the strategy Title & Date we retrieve only 1 search result from YouTube. For this single search result, we retrieve its 50 related videos. For these 50 related videos we find the most similar item in the Open Images dataset and for the final scoring function we use the log-smoothing function.

5.3.3 Strategy Title & Domain

Finally, we tuned the parameters of the strategy Title & Domain. We started with tuning the parameter search results. Again, it is best to retrieve only 1 search result, see table 5.6. It is also the best to retrieve 50 related videos, although this can not be seen from the table. This is because of a rounding error. The MAP-score for retrieving 50 related videos is 0.0661 while it is 0.0660 for retrieving 5 related videos. It is best to retrieve only 1 Open Images video for a given YouTube video. The only difference with the other strategies is that it is best to use the no-log-smoothing function for this strategy. It seems to be coincidence that in this case the best related videos are ranked higher and therefore it is better to punish the lower ranked videos more to filter the noise. However more research is needed to find out if this is indeed the case.

So, to summarize, for the strategy Title & Domain we retrieve only 1 search result from YouTube. For this single search result, we retrieve its 50 related videos. For these 50 related videos we find the most similar item in the Open Images dataset and for the final scoring function we use the no-log-smoothing function.

Parameter	Value 1	Value 2	Value 3
Number of search results (50/20/1)	0.01	0.03	0.07
Number of related videos (50/5/1)	0.07	0.07	0.00
Number of similar Open Images videos given a YouTube video (5/1)	0.03	0.07	-
Scoring function (no-smoothing/smoothing)	0.07	0.01	-

Table 5.6: Mean Average Precision (MAP) for the different values of the different parameters

Chapter 6

Case study: user study and results

This chapter discusses the set up of the evaluation of the proposed model and the results of this evaluation. In history most often only accuracy was measured to evaluate the performance of a recommendation algorithm. However, according to [46] [29] [41] [22], the recommendations that are most accurate are sometimes not the ones that are most useful to users. Therefore it is better to measure users' satisfaction towards recommendations.

This could be measured in a user study. However, a drawback of a user study is that they are more expensive to perform than an offline experiment. According to [41], an offline experiment could be used to filter out inappropriate algorithms, leaving a relatively small set of candidate algorithms to be tested by the more costly user studies. A typical example of this process is when the parameters of the algorithms are tuned in an offline experiment, and then the algorithm with the best tuned parameters continues to the next phase.

For each of the three proposed strategies from the previous chapter the parameters were tuned in an offline experiment. These strategies are compared to tag-based recommendations and random recommendations in a user study. The set up of this user study is discussed in section 6.1. Finally, we discuss the results that follow from this user study in section 6.2.

6.1 Experimental setup

As a follow up of the offline experiments, either a user study or online experiment could have been performed. However, it was not possible to make an experiment using the live website that is managed by Sound & Vision. Therefore we performed a user study.

In the user study, users' satisfaction towards recommendations from a newly bootstrapped domain, generated by different algorithms, is measured. The goal of the experiment is to find an answer to the question if the model researched in this thesis is an improvement over random recommendations and tag-based recommendations. Therefore the best performing algorithms from the offline experiment are compared to these baselines. This leads to the conceptual model shown in figure 6.1.

Different methods exist to measure the performance of the algorithms and compare them to each other. In this thesis the pairwise comparison method is chosen. The



Figure 6.1: Conceptual model

advantage of this method is that it is easier for a participant to say what she prefers than to give a rating on a scale. Next, the advantage of using pairwise comparison over rank order is that it also quantifies the difference of preferences among the algorithms. The disadvantage is that it only measures which algorithm is preferred, but not how good the most preferred algorithm is performing and also not why people prefer one algorithm over the other. Furthermore it also does not measure how much a participant prefers an algorithm. It might be that more people prefer algorithm A over B, but algorithm A is in most cases just slightly preferred over B, while in some other cases B is more preferred over A. Incorporating scores in the pairwise comparison to avoid this also brings its own problems, like the difficulty for a participant to give a score on a scale, and we decided not to do this.

We chose a within-subjects design to set up this pairwise comparison experiment. The advantage is its increased statistical power and therefore fewer participants are needed. The disadvantage is that within-subjects designs are subject to order effects, like practice effects and fatigue effects. To avoid these effects, counterbalancing is applied, which means that the order in which the pairs are presented to a participant is randomized.

In the rest of this section, first we shortly refresh the strategies that are compared and on which domains they are evaluated. This is discussed in section 6.1.1. Next, we introduce the task that the participants had to do in section 6.1.2. Finally, we discuss the threats of this set up in section 6.1.3.

6.1.1 Evaluated strategies and domains

In the user study we compared 5 strategies: Title, Title & Date, Title & Domain, Random and Tags. The first three strategies are an implementation of the proposed model and are introduced in chapter 5. The parameters of these strategies are tuned in an offline experiment that is discussed in the previous chapter.

Instead of using the best configurations that resulted from the offline experiment, we decided to use 50 search results instead of 1 search result as configuration for the model that is used in the user study. As using 1 search result works better according to the offline experiment we expect that the results that would have followed from the

user study using 1 search result are better than the results that came out of the user study with the current configuration.

The other strategies are Random and Tags. The strategy Random just randomly selects 5 recommendations, while the strategy Tags uses the tags that we retrieved together with the Open Images dataset. The recommended videos are the ones from another domain that share the most tags with the source video.

These strategies are evaluated using videos from different domains. As only the domains news item, documentary/report and event coverage contain a substantial amount of videos, only videos from these domains are selected. For the user study, the same 5 randomly selected videos from each domain that were used in the offline experiment are used. For the domains news item and event coverage, cross-domain recommendations are given from the domain documentary/report. For the domain documentary/report, cross-domain recommendations are given from the domain news item. This is the same as we did in the offline experiment.

6.1.2 Task

The strategies are compared using a pairwise comparison experiment. This is implemented in the task shown in figure 6.2.

In the task, the participants are shown a video from a certain domain. For this video, the title and domain are given. Next to this video two top-5 lists of recommendations are shown. For each recommended item, a thumbnail, title and domain are given. The recommendations are from another domain and computed by different strategies. The task for the participant is to choose the list which is most satisfactory for her, given that she likes the given video and now want to explore the newly bootstrapped domain. It was assumed that participants were able to make their decision based on the given information, as the same information for the recommendations (without domain) is given in real life applications such as YouTube. The instruction text from table 6.1 was given before the experiment. We did this to make the participants understand that they had to choose the most satisfactory recommendation list given that they like the video and not choose the recommendation list they like most (without taking the given video into account).

After this instruction text, we collected the biographical data of the participants to make sure that the results were not biased towards a certain group. Their name, age and occupation was asked, telling them that this information would only be used for academical purposes. The reason that we collected their names is that we would be able to contact the participant afterward if we wanted to have an explanation for her choices.

So, after the instruction text and the collection of biographical data, the participant was shown one of the selected videos. For this video, we compared 5 different strategies. We created a full combination of these strategies resulting in 20 comparisons per video. To give an example, we tested both the pair Title vs. Random and Random vs. Title. The order in which these pairs were presented to the participant was randomized. After finishing 20 comparisons the participant was allowed to comment on the experiment and submit her answers. We made the decision to allow the participant to only submit the answers after 20 comparisons as we had seen earlier in the annotation task that the participants solved only a small number of tasks. To encourage the par-



Figure 6.2: Overview of the pairwise comparison task

participants to fulfill the experiment we showed a counter so that they knew how far they were with the experiment.

For each video we asked multiple participants about their opinion to increase the reliability of the results. We decided to ask acquaintances to take part in the experiment, so that it would be quite easy to ask more information from the participants afterward if needed as not every participant will make use of the option to comment on the experiment. As acquaintances are seen as trusted participants, we decided to

Intro

Imagine that you are surfing to a video website. You find a video that you appreciate and click on it. You watch the complete video. The video you just watched was a news item about gun usage in the United States. Now you have seen enough news items and want to explore documentaries. The recommender system recommends you documentaries it thinks you will appreciate given that you also appreciate a news item about gun usage in the United States. Such a recommendation might be the documentary *Bowling for Columbine*.

Your task

This is the kind of task you have to solve. First you will see a video which is from a certain domain. Which domain that is will be given in the task. This is for example news items. Then you will see two lists of recommendations, for example from the domain documentaries. Again, this will be given in the task. Your task is to *choose which of the two recommended lists you would prefer, given that you appreciate the shown video from the given domain and that you now want to explore the other domain*. So if you don't like the videos, try to imagine that you are a person that likes the given video and don't take your own preferences too much into account. For example, if a video about fires is shown and a video about football is recommended, this might be a bad recommendation in your eyes, even when you like football. Furthermore, it can happen that there are no videos to recommend. Then you will be shown a list with no videos or maybe a shorter list.

Length of the experiment

In total, you will be shown *20 combinations* after which you can submit. It is also possible to give comments then about the complete test. After 20 comparisons you can decide to do another video and move on, just click start experiment for that. Your personal information will be asked again in that case.

Table 6.1: Instruction text of the user study

ask only 3 participants per video. As none of the participants is an expert on the Open Images dataset we treated all participants in the same way.

The whole experiment was set up using the BruteForce platform, also used for the manual annotation task to annotate domains. In total 15 videos were evaluated by 3 participants. For each video 20 comparisons were made. So, 900 data points were collected in total.

6.1.3 Threats to validity

There are a lot of threats to the validity of the experiment. One of them is that the participants are choosing the recommendations they like most based on their personal preference, without taking the given video into account. We tried to avoid this by giving a clear instruction text.

Other factors that can influence the preference of the participant are the language of the shown metadata (i.e. title and domain) and the quality of the thumbnails. Therefore

these are kept constant in the user study where possible. It was not possible to keep the color of the thumbnails constant as some of the videos were black/white and some were in color. However, some participants might find it less satisfactory to receive black/white videos as recommendations for a colored video. So, this is part of what we want to measure and is not seen as a threat.

Finally, some algorithms were unable to recommend 5 videos for some videos. In those cases less videos were recommended. This is also a factor that might influence the preference of the participant. However, participants might find it less satisfactory to receive less recommendations. Therefore this is part of what we want to measure and is not seen as a threat.

6.2 Results of the user study

In this section the results of the user study are given. First some general information about the user study and the participants is given in section 6.2.1. Next, the results of the comparisons made are given in section 6.2.2. Using these results we can answer the question which strategy is preferred by the participants on this dataset. However these results do not give insight in why these results occur. Therefore why did a small analysis to be able to have a first explanation about what is happening. This analysis is discussed in section 6.2.3.

6.2.1 General information about the user study and participants

The user study ran for 17 days. In this period 21 persons solved all comparisons. On average 2.14 videos (sd: 1.42) were solved per participant. The average age of these participants is 27.24 years (sd: 10.03). 71.4% of the participants is student and two third of these students study computer science. Finally, we also collected comments about the experiment which are given in appendix G.

6.2.2 Comparing strategies

For each tested video, three different participants solved each 20 comparisons. The total number of votes that each strategy got for the different videos is presented in table 6.2.

We tested each pair of strategies twice for each video by switching sides. So, we tested both the pair Title vs. Random and Random vs. Title. If each participant would constantly have chosen the left or right option or alternate between the two, each strategy should have the same number of votes. Next, the total number of votes for each strategies should be the same if all participants would have chosen their preference randomly. However, there could be some deviation when everything would have been chosen randomly. We think that there would be a deviation of maximally 10% if everything is chosen randomly. In that case, all total number of votes should be between 162 and 198. As this is not the case, and it is also not the case that all strategies have the same number of votes, we conclude that the results reflect the preference of the participants for the different strategies.

Next we want to answer the question which strategy is preferred most. Therefore we want to create a ranking. We can do that for each video by giving the strategy with

Video	Title	Title & Date	Title & Domain	Random	Tags
News item 1	13	16	14	9	8
News item 2	10	11	10	14	15
News item 3	3	17	7	23	10
News item 4	12	8	14	13	13
News item 5	9	20	12	5	14
Documentary/report 1	14	11	4	7	24
Documentary/report 2	10	14	21	5	10
Documentary/report 3	10	5	17	4	24
Documentary/report 4	17	12	12	10	9
Documentary/report 5	20	5	10	9	16
Event coverage 1	18	11	8	0	23
Event coverage 2	3	7	11	18	21
Event coverage 3	17	15	17	5	6
Event coverage 4	14	8	6	14	18
Event coverage 5	3	8	18	8	23
Total	173	168	181	144	234

Table 6.2: The number of votes for each strategy for each video

the most votes a score of 1 and the strategy with the least votes a score of 5. Equal number of votes will get the same score. To create an absolute rank from the partial ranks, we sum up the scores for each strategy, creating an absolute rank. These results are presented in table 6.3.

Video	Title	Title & Date	Title & Domain	Random	Tags
News item 1	3	1	2	4	5
News item 2	4	3	4	2	1
News item 3	5	2	4	1	3
News item 4	4	5	1	2	2
News item 5	4	1	3	5	2
Documentary/report 1	2	3	5	4	1
Documentary/report 2	3	2	1	5	3
Documentary/report 3	3	4	2	5	1
Documentary/report 4	1	2	2	4	5
Documentary/report 5	1	5	3	4	2
Event coverage 1	5	2	3	4	1
Event coverage 2	5	4	3	2	1
Event coverage 3	1	3	1	5	4
Event coverage 4	2	4	5	2	1
Event coverage 5	5	3	2	3	1
Total	48	44	41	52	33

Table 6.3: Partial and absolute ranking for the different strategies

It can be seen that the Tags strategy is the most preferred, followed by the Title &

Domain, Title & Date, Title and Random strategies. This differs from the ranking that could be created based on the total number of votes from table 6.2. The 6 ties are the cause of this difference.

At least, from both rankings we can conclude that the Tags strategy is preferred most often by far. The preference of the three different strategies of the model is closer to each other. However, at least they all outperform the Random strategy.

To create a final assessment we took a look at the performance of the different strategies compared to each other. Therefore we aggregated the results based on pairs of strategies. The aggregated results are presented in table 6.4. As the two versions of a pair, e.g. 1-2 and 2-1, is actually the same comparison, we combined those results together. The inconsistency column in the table shows the number of times that a participant was inconsistent in choosing between two versions of a pair.

1st strategy	2nd strategy	#Votes 1st	#Votes 2nd	# Inconsistent
1	2	21	15	9
1	3	16	22	7
1	4	24	17	4
1	5	13	27	5
2	3	19	19	7
2	4	19	13	13
2	5	15	27	3
3	4	25	17	3
3	5	14	27	4
4	5	13	28	4

Table 6.4: Number of times a strategy is preferred over another strategy. 1 = Title, 2 = Title & Date, 3 = Title & Domain, 4 = Random, 5 = Tags

Again, the Tags strategy outperforms all other strategies. Also, all strategies outperform the Random strategy. This is in line with the earlier conclusion. Furthermore, the Title strategy is preferred over the Title & Date strategy, the Title & Domain strategy is preferred over the Title strategy and none of the Title & Date and Title & Domain strategy is preferred. This seems to be inconsistent and is in line with the earlier conclusion that the preference for the different strategies of the model is close to each other.

Looking at inconsistency, in 59 of the 450 cases (13.11%) a participant was inconsistent. We think that this is quite often and together with the earlier finding that participants did not choose randomly or by a pattern, we conclude that it was difficult for the participants to choose between two lists of recommendations. This is in line with the gathered comments presented in appendix G.

For some comparisons it seemed to be impossible to choose the best recommendation as multiple participants were inconsistent on that comparison. That happened for 7 comparisons. Furthermore it can be noticed that participants were more often inconsistent when choosing between the strategies Title & Date and Random, than when comparing other strategies. Apparently the Title & Date strategy does not perform that much better than Random, which is in line with the total number of votes from table 6.2.

Next we used majority voting to decide on the best strategy for each pair of strategies for each video. To create a majority voting we decided that each participant could have only one preference or is inconsistent. In cases where two participants are inconsistent we decided that no majority could be created, as this would actually mean that one participant decides on the majority. The results of this majority voting is presented in appendix F.

Finally we took a look at the performance on the domain level. Using the numbers from table 6.2, we made the rankings on domain level. The results are shown in table 6.5.

News item	Documentary/Report	Event coverage
Title & Date (72)	Tags (83)	Tags (91)
Random (64)	Title (71)	Title & Domain (60)
Tags (60)	Title & Domain (64)	Title (55)
Title & Domain (57)	Title & Date (47)	Title & Date (49)
Title (47)	Random (35)	Random (45)

Table 6.5: Strategy ranking on domain level with the number of votes between brackets

The rankings for the domains Documentary/Report and Event coverage are almost the same and in line with the earlier results. The ranking for the domain news item is quite different. However, it can be seen that the total number of votes for the different strategies is quite close to each other and the final ranking seems to be more coincidence than that there is a significant difference in the preference for a strategy for that domain.

6.2.3 Towards an explanation for the results

To explain the given results, first we took a look at when a strategy could not compute recommendations for a video. This is the case for the Tags strategy for the video news item 1 and for the video documentary/report 4. The strategy Title & Domain could not compute recommendations for the video event coverage 4 as YouTube did not return a search result for that video. It can be seen in appendix F that in these cases the strategy that produces no recommendations is never preferred by a majority. However, in table 6.2 we can see that those strategies still got some votes for those videos. For the videos news item 1 and event coverage 4, only one participant preferred the no recommendations option. As the other two participants were consistent on the other option, it had no influence on which strategy was preferred using majority voting.

For the video documentary/report 4, one participant always preferred the no recommendation option, creating 8 votes and one participant was inconsistent in choosing between Random and Tags, voting once for the Tags, and thus no recommendations, strategy. To understand the reason why participants preferred the no recommendation option we contacted them, which was possible since we collected the names of the participants. All participants commented that they chose the no recommendation option because the other option was very bad.

So, for the videos news item 1, documentary/report 4 and event coverage 4, at least one participant thinks that the proposed model is working so bad that she prefers

to get no recommendations. To understand why it is working so bad, we took a look at when the exact same video is found as first search result on YouTube using one of the strategies. These results are presented in table 6.6.

Video	Title	Title & Date	Title & Domain
News item 1	No	No	No
News item 2	Yes	Yes	No
News item 3	Yes	Yes	No
News item 4	No	No	No
News item 5	Yes	Yes	No
Documentary/report 1	Yes	No	No
Documentary/report 2	Yes	Yes	No
Documentary/report 3	Yes	Yes	No
Documentary/report 4	No	No	No
Documentary/report 5	Yes	Yes	No
Event coverage 1	No	No	No
Event coverage 2	No	No	No
Event coverage 3	No	Yes	No
Event coverage 4	Yes	No	No
Event coverage 5	Yes	No	No

Table 6.6: Using the different strategies, can we find the same video on YouTube as first search result?

In the cases of the videos news item 1 and documentary/report 4, the found YouTube video is different from the Open Images video. Apparently this creates bad recommendations. However, for the video event coverage 4, the strategy Title finds the same video on YouTube. Therefore we took a look at the related videos for this video.

For the video event coverage 4 (title: "Jaarlijkse Vondelpark estafette"), we received 23 search results using the strategy Title. In total there are 772 related videos. The first search result, which is the exact same movie, has 25 related videos. Only 2 of those 25 videos are not about the "Vondelpark". This leads to the hypothesis that the relations on YouTube are content-based for this video.

Next we took a look at the recommendations and why they were created. The recommendations are given in table 6.7.

"Heidekoningin ter schapenmarkt Ede"
"Hondenvoetbal"
"Keizer bolling met de krulbol"
"De wereld rond met een rollende ton"
"Nieuwste caravans en tenten op RAI-tentoonstelling in Amsterdam"

Table 6.7: Recommendations for the video "Jaarlijkse Vondelpark estafette" using the Title strategy

As an example to show what is happening inside our model, we explain the recommendation "Hondenvoetbal". The 18th search result of the video "Jaarlijkse Vondelpark estafette" is the video "Hondenreunie in Vondelpark (1927)". Most of the related

videos for this video are about dogs (Dutch: honden) and the most similar Open Images video is "Hondenvoetbal" for those videos. As we sum up all the scores, this video gets a high overall score. So this recommendation is actually a result of noise incorporated in the model which makes the strategy perform like the Random strategy. This is one explanation for the fact that there is no relation between the preference for a strategy and the strategy being able to find the same video on YouTube.

However, one of the related videos for the first search result is the video "Vondelpark 100 jaar", which is also in the Open Images dataset. Because the video "Hondenvoetbal" is more often linked to a related video, it gets a higher score. It can be questioned if that improves the model.

Finally we took a look at the recommendations when we would have used only 1 search result. In that case the recommendations would have been: "Beelden in Vondelpark", "Keizer bolling met de krulbol" and "Wereldreiziger". At least the first recommendation would be more satisfying than the recommendations computed using 50 search results. However, the drawback is that using only 1 search result creates less recommendations.

Next, as an example we took a look at the recommendations for the video documentary/report 5 (title: "Een wonderlijke hobby"), which is about someone collecting scrap. The recommendations for the different strategies are listed in appendix H. The recommendations for the Tags strategy are based on the shared tag "Hobby's en verzamelingen". From the votes displayed in the table we can see that the Title and Tags strategy are most preferred. It is assumed that this is the case because of the recommended video "Kampeertoonstelling" by the Title strategy and the video "Nationale ruilbeurs verzamelaars" for the Tags strategy, as these videos seem to be the closest to someone collecting scrap and showing it. However it is just guessing why the participants preferred a strategy. That is partly because it seems that all videos are not very good recommendations. This is in line with the ground truth created in chapter 5 which says there is only one relevant item for the video documentary/report 5. This makes it hard for the participants to decide which strategy they prefer which is in line with earlier findings.

A final observation that we could make from table 6.6 is that it is best to use only the title if we want to find the same video on YouTube. Combined with an earlier observation that using the title as input returns a search result most often, we conclude that the Title strategy is the best strategy to use for our model for the Open Images dataset. The reason that this does not follow from the user study can be explained by the fact that on this dataset it was hard to create a difference between the strategies.

Chapter 7

Conclusions and Future Work

In this chapter we conclude our work and describe future work. First in section 7.1 we will explain how we addressed the research questions posed in chapter 1. Next, a discussion and reflection on our work is given in section 7.2. Finally, the future work is discussed in section 7.3.

7.1 Conclusions

RQ1 Which are the strengths and limitations of the current approaches to bootstrap new domains in cross-domain recommendations?

In order to answer this research question we did a literature study to create an overview of the current approaches. We found that currently the problem addressed in this thesis is solved using tags or Semantic Web based solutions. Next, we tried to understand the working of these approaches and the strengths and limitations. We found that the biggest weakness of these solutions is that they are content-based. This is a weakness because only items that share a content-based feature can be related using that methods. However, also items that do not share content-based features might be related to each other in real life. Therefore we proposed a new model that uses users' preferences to create relations between a new domain and existing domains in a cross-domain recommender system. These users' preferences are transferred from an auxiliary system from the Social Web.

RQ2 What is the best way to evaluate our proposed model?

In order to answer this research question we did a literature study on evaluating recommender systems. We found that the best way to evaluate a recommender system is to filter out inappropriate algorithms using an offline experiment and to evaluate the best algorithms in a user study or online experiment. Therefore we used an offline experiment to tune the parameters in our model for three different strategies. These three strategies were compared to two baseline methods in a user study.

RQ3 What is the best configuration of our proposed model?

In order to answer this research question we set up a case study. We selected the Open Images dataset and performed a manual annotation task to split up the dataset into subsets where videos were annotated according to the following definition of domain: "A domain is the name of a group of items characterized by its video type". In total 193 videos could be annotated with one domain. These videos were used for the rest of the case study.

Next, we selected YouTube as auxiliary system. We set up a couple of experiments to explore how YouTube works and found that YouTube is not performing as most information retrieval systems do. For example, it matters if the singular or plural form of a word is used to search videos. We used this understanding to guide us during the rest of the case study.

Next, we created a ground truth as we wanted to tune the parameters of our model in an offline experiment. This ground truth contains relevant judgments for cross-domain recommendations for multiple randomly selected videos. We proposed a definition of relevance for the Open Images dataset in order to make this possible. We found that for each selected video only a few other cross-domain videos are relevant.

Next, we selected Title, Title & Date and Title & Domain as the three strategies to implement our model. The difference between the strategies is the input used to query the YouTube dataset. We found that the strategy Title most often got back a result using the search functionality in the YouTube API. For a given Open Images video, the strategy Title also found most often the same video on YouTube compared to the other strategies.

Finally, we tuned the parameters for each strategy using the created ground truth and the Average Precision metric. We found that on this dataset, using the created ground truth, the best thing is to use only the first video that could be retrieved using the search functionality of the YouTube API. We also found that it is best to retrieve all 50 related videos, which is the maximum, for the retrieved video using the YouTube API. Next, we found that for each of these related videos it is best to find only one most similar Open Images video. Finally we found that for the strategy Title and Title & Date it is the best thing to use the proposed log-smoothing-function to compute scores and for the strategy Title & Domain it is best to use the no-log-smoothing-function to compute scores. These scoring functions returned a score and the 5 cross-domain videos with the highest score were recommended.

RQ4 How well does our proposed model work compared to other approaches?

In order to answer this question, we set up a user study to compare the users' satisfaction towards the recommendations created by the three strategies of our model, towards tag-based recommendations and towards random recommendations. Instead of using the best strategies that came out of the offline experiment, we used a configuration of our model that uses 50 search results instead of 1, as we wanted to see if that would result in recommendations that are observed as less satisfying. Using this configuration we found using a pairwise comparison that the tag-based recommendations were preferred. The recommendations created by our model were only slightly preferred over the random recommendations.

RQ What is the potential of a Social Web based solution to bootstrap new domains

in cross-domain recommendations?

In order to answer the main research question, we answered all sub-questions. Next, we also performed an analysis to understand the working of the model and the outcomes of the user study. This gave us interesting insights about the potential of the proposed solution.

First we found that the dataset needs to contain videos that are watched quite often when YouTube is used as auxiliary system to increase the chance of having related videos based on users' preferences. We found in literature that videos are related if they are co-watched in one session in the last 24 hours and content-based otherwise. Due to the videos in the Open Images that are not being popular, we found that the related videos were content-based, which confirms what we found in literature.

Next, we found that our model works best when only 1 search result is used. Using more search results incorporates too much noise in the model, which leads to weak recommendations. The drawback of using only 1 search result is that in general less recommendations can be given than when multiple search results are used.

Finally, we found that the related videos should be quite similar to the videos in the source system. Otherwise the final strength of the relation between an item from the new domain and an item from an existing domain is very weak, which results in weak recommendations.

As the Open Images dataset does not fulfill the requirements above, the proposed model does not have potential for the Open Images dataset using YouTube as auxiliary system.

7.2 Discussion/Reflection

In this section we will discuss and reflect on our work.

First, we made the decision to use the Open Images dataset for our case study. The advantage was that we could also retrieve tags for this dataset, but the disadvantage was that it did not contain multiple domains. Therefore we had to set up a manual annotation task to annotate the dataset with domains.

Second, we made the decision to do a manual annotation task to split up this dataset into domains. We tried to make the instructions as clear as possible for the users. However, it was not for everyone that clear when a video was a news item and it was an event coverage. This created some noise in the final dataset, slightly influencing the end results.

Third, the ground truth created is fully based on our proposed definition. In general a ground truth could be created from historical data or users' ratings. That would be more reliable. However, that was not possible in this case. The only improvement that we could have made is to do an interview to ask people when they think something is relevant. That would also create more insight in why people prefer certain recommendations.

Fourth, we did an offline experiment and user study. The dataset made it difficult for the participants of the user study to decide which strategy they preferred as in general there was only one or no relevant recommendation.

Finally, we did an analysis to better understand the working of the model. This created insight in the potential of the proposed model on the chosen dataset and created a good basis for future work.

7.3 Future work

Earlier we concluded that our proposed model has no potential on the Open Images dataset using YouTube as auxiliary system. But we found indications that our model might work under different circumstances. In this section we discuss future work that might help us answer the general question if the proposed model has potential.

First, our proposed model needs to be evaluated on other datasets to claim generalization of the obtained results. As we found that our models works the best when the videos in the source dataset are the same as the videos in the auxiliary dataset, we recommend to use a big dataset in future work. Furthermore the dataset needs to contain videos that are watched quite often when YouTube is used as auxiliary system to increase the chance of having related videos based on users' preferences instead of content-based relations. And finally a different auxiliary system can be used to evaluate the performance of the proposed model.

Second, the model itself can be improved at some points. The strength of a YouTube search result can be computed, e.g. using Lucene. In this way the final score can be better computed. Furthermore, the computed strength can be used to decide to do another query using a different input. Next, thresholds can be used to filter out inappropriate recommendations. This will lead to less recommendations, but will not show bad recommendations anymore. And finally, different scoring functions can be used to tweak the final recommendations.

Third, our proposed model can be compared to a Semantic Web based solution. This will give a better indication of the potential of our model as tags have their drawbacks to bootstrap new domains in cross-domain recommendations as well.

Finally, to be able to give good recommendations it is important to understand the user and her information needs. Therefore we recommend to research why people prefer certain recommendations. Using this information the proposed model could possibly be adapted or it could be concluded that the proposed model has no potential at all. Part of this research can be to research which domains are close together and why people want to get cross-domain recommendations.

Bibliography

- [1] Fabian Abel, Eelco Herder, Geert-Jan Houben, Nicola Henze, and Daniel Krause. Cross-system user modeling and personalization on the Social Web. *User Modeling and User-Adapted Interaction*, 23(2-3):169–209, November 2012.
- [2] G. Adomavicius and A. Tuzhilin. Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering*, 17(6), 2005.
- [3] Hyung Jun Ahn. A new similarity measure for collaborative filtering to alleviate the new user cold-starting problem. *Information Sciences*, 178:37–51, 2008.
- [4] Daniel Appelquist, Dan Brickley, Melvin Carvahlo, Renato Iannella, Alexandre Passant, Christine Perey, and Henry Story. A standards-based, open and privacy-aware social web, 2010.
- [5] Marko Balabanović and Yoav Shoham. Fab: content-based, collaborative recommendation, 1997.
- [6] Tim Berners-Lee, James Hendler, and Ora Lassila. The semantic web. *Scientific American*, 284(5):34–43, May 2001.
- [7] Daniel Billsus, Michael J. Pazzani, and James Chen. A learning agent for wireless news access. In *Proceedings of the 5th International Conference on Intelligent User Interfaces*, IUI '00, pages 33–36, New York, NY, USA, 2000. ACM.
- [8] JS Breese and D Heckerman. Empirical analysis of predictive algorithms for collaborative filtering. *Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence*, pages 43–52, 1998.
- [9] Robin Burke. Hybrid Recommender Systems : Survey and Experiments. *User Modeling and UserAdapted Interaction*, 12(4):331–370, 2002.
- [10] Paolo Cremonesi, Antonio Tripodi, and Roberto Turrin. Cross-Domain Recommender Systems. *2011 IEEE 11th International Conference on Data Mining Workshops*, pages 496–503, December 2011.

- [11] James Davidson, Blake Livingston, Dasarathi Sampath, Benjamin Liebald, Junning Liu, Palash Nandy, Taylor Van Vleet, Ullas Gargi, Sujoy Gupta, Yu He, and Mike Lambert. The YouTube video recommendation system. *Proceedings of the fourth ACM conference on Recommender systems - RecSys '10*, page 293, 2010.
- [12] Marco Degenmis, Pasquale Lops, and Giovanni Semeraro. A content-collaborative recommender that exploits WordNet-based user profiles for neighborhood formation, 2007.
- [13] Ignacio Fernández-Tobías, Iván Cantador, Marius Kaminskas, and Francesco Ricci. A generic semantic-based framework for cross-domain recommendation. *Proceedings of the 2nd International Workshop on Information Heterogeneity and Fusion in Recommender Systems - HetRec '11*, pages 25–32, 2011.
- [14] Ignacio Fernández-Tobías, Iván Cantador, Marius Kaminskas, and Francesco Ricci. Cross-domain recommender systems : A survey of the State of the Art. *Proceedings of the 2nd Spanish Conference on Information Retrieval. CERI, 2012*.
- [15] David Goldberg, David Nichols, Brian M. Oki, and Douglas Terry. Using collaborative filtering to weave an information tapestry, 1992.
- [16] Gustavo González, Beatriz López, Josep Lluís, and De Rosa. A Multi-agent Smart User Model for Cross-domain Recommender Systems. *Proceedings of Beyond Personalization, 2005*.
- [17] Jonathan L. Herlocker, Joseph A. Konstan, Loren G. Terveen, and John T. Riedl. Evaluating collaborative filtering recommender systems, 2004.
- [18] W. Hill, L. Stead, M. Rosenstein, and G. Furnas. Recommending and evaluating choices in a virtual community of use. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 194–201. ACM Press/Addison-Wesley Publishing Co. New York, NY, USA, 1995.
- [19] Marius Kaminskas and Francesco Ricci. Location-adapted music recommendation using tags. In *User Modeling, Adaption and Personalization*, pages 183–194. Springer, 2011.
- [20] Bart P. Knijnenburg, Martijn C. Willemsen, Zeno Gantner, Hakan Soncu, and Chris Newell. Explaining the user experience of recommender systems, 2012.
- [21] Joseph A. Konstan, Bradley N. Miller, David Maltz, Jonathan L. Herlocker, Lee R. Gordon, and John Riedl. GroupLens: applying collaborative filtering to Usenet news, 1997.
- [22] Joseph a. Konstan and John Riedl. Recommender systems: from algorithms to user experience. *User Modeling and User-Adapted Interaction*, 22(1-2):101–123, March 2012.

- [23] Xuan Nhat Lam, Thuc Vu, Trong Duc Le, and Anh Duc Duong. Addressing cold-start problem in recommendation systems. *Proceedings of the 2nd international conference on Ubiquitous information management and communication - ICUIMC '08*, page 208, 2008.
- [24] Bin Li. Cross-Domain Collaborative Filtering: A Brief Survey. *2011 IEEE 23rd International Conference on Tools with Artificial Intelligence*, pages 1085–1086, November 2011.
- [25] Bin Li, Qiang Yang, and Xiangyang Xue. Can movies and books collaborate? - crossdomain collaborative filtering for sparsity reduction. In *International Joint Conference on Artificial Intelligence*, pages 2052–2057, 2009.
- [26] Antonis Loizou. *How to recommend music to film buffs: enabling the provision of recommendations from multiple domains*. PhD thesis, University of Southampton, 2009.
- [27] Pasquale Lops, Marco De Gemmis, and Giovanni Semeraro. Content-based Recommender Systems: State of the Art and Trends. In Francesco Ricci, Lior Rokach, Bracha Shapira, and Paul B Kantor, editors, *Recommender Systems Handbook*, chapter 3, pages 73–105. Springer US, 2011.
- [28] Christopher D. Manning and Prabhakar Raghavan. *An Introduction to Information Retrieval*, 2009.
- [29] S M McNee, J Riedl, and J a Konstan. Being accurate is not enough: how accuracy metrics have hurt recommender systems. *CHI'06 extended abstracts on Human factors in computing systems*, page 1101, 2006.
- [30] Seung-taek Park. Pairwise Preference Regression for Cold-start Recommendation Categories. *Methodology*, 37:21–28, 2009.
- [31] Michael Pazzani and Daniel Billsus. Learning and Revising User Profiles: The Identification of Interesting Web Sites. *Machine Learning* 27, 331:313–331, 1997.
- [32] Michael J. Pazzani and Daniel Billsus. Content-Based Recommendation Systems. *The Adaptive Web*, 4321:325–341, 2007.
- [33] Pearl Pu, Li Chen, and Rong Hu. A user-centric evaluation framework for recommender systems. In *Proceedings of the fifth ACM conference on Recommender systems - RecSys '11*, page 157, 2011.
- [34] Pearl Pu, Li Chen, and Rong Hu. Evaluating recommender systems from the user's perspective: survey of the state of the art. *User Modeling and User-Adapted Interaction*, 22(4-5):317–355, March 2012.
- [35] Paul Resnick, Neophytos Iacovou, Mitesh Suchak, Peter Bergstrom, and John Riedl. GroupLens: An Open Architecture for Collaborative Filtering of Netnews. In *Proceedings of the 1994 ACM conference on Computer supported cooperative work - CSCW '94*, pages 175–186. ACM Press, 1994.

- [36] C. Robson. *Real World Research - A Resource for Social Scientists and Practitioner-Researchers*. Blackwell Publishing, Malden, second edition, 2002.
- [37] Shaghayegh Sahebi and William W Cohen. Community-Based Recommendations : a Solution to the Cold Start Problem. In *Workshop on Recommender Systems and the Social Web (RSWEB), held in conjunction with ACM RecSys'11*, Chicago, 2011.
- [38] Badrul Sarwar, George Karypis, Joseph Konstan, and John Reidl. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the tenth international conference on World Wide Web - WWW '01*, pages 285–295, New York, New York, USA, 2001. ACM Press.
- [39] J Ben Schafer, Dan Frankowski, Jon Herlocker, and Shilad Sen. Collaborative Filtering Recommender Systems. In *The Adaptive Web*, chapter Collaborat, pages 291–324. Springer-Verlag, Berlin, Heidelberg, 2007.
- [40] Andrew I. Schein, Alexandrin Popescul, Lyle H. Ungar, and David M. Pennock. Methods and metrics for cold-start recommendations. *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval - SIGIR '02*, page 253, 2002.
- [41] Guy Shani and Asela Gunawardana. Evaluating Recommendation Systems. In *Recommender Systems Handbook*, pages 257–297. Springer, 2011.
- [42] Bracha Shapira, Lior Rokach, and Shirley Freilikhman. Facebook single and cross domain data for recommendation systems. *User Modeling and User-Adapted Interaction*, 23(2-3):211–247, September 2012.
- [43] Upendra Shardanand and Pattie Maes. Social information filtering. In *Proceedings of the SIGCHI conference on Human factors in computing systems - CHI '95*, pages 210–217. ACM Press, 1995.
- [44] Yue Shi, Martha Larson, and Alan Hanjalic. Tags as bridges between domains: Improving recommendation with tag-induced cross-domain collaborative filtering. *User Modeling, Adaption and Personalization*, pages 305–316, 2011.
- [45] Rashmi Sinha and Kirsten Swearingen. Comparing Recommendations Made by Online Systems and Friends. In *In Proceedings of the DELOS-NSF Workshop on Personalization and Recommender Systems in Digital Libraries*, 2001.
- [46] Xiaoyuan Su and Taghi M. Khoshgoftaar. A Survey of Collaborative Filtering Techniques. *Advances in Artificial Intelligence*, 2009:1–19, 2009.
- [47] Kirsten Swearingen and Rashmi Sinha. Beyond algorithms: An hci perspective on recommender systems. In *ACM SIGIR. Workshop on Recommender Systems*, volume Vol. 13, Numbers 5-6, pages 393–408, 2001.
- [48] Martin Szomszor, Harith Alani, Ivan Cantador, Kieron O Hara, and Nigel Shadbolt. Semantic Modelling of User Interests Based on Cross-Folksonomy Analysis. In *Proceedings of the 7th International Semantic Web Conference (ISWC 2008)*, pages 632–648, Karlsruhe, Germany, 2008.

-
- [49] Pinata Winoto and Tiffany Tang. If You Like the Devil Wears Prada the Book, Will You also Enjoy the Devil Wears Prada the Movie? A Study of Cross-Domain Recommendations. *New Generation Computing*, 26(3):209–225, June 2008.
- [50] Zi-Ke Zhang, Chuang Liu, Yi-Cheng Zhang, and Tao Zhou. Solving the cold-start problem in recommender systems with social tags. *EPL (Europhysics Letters)*, 92(2):28002, October 2010.
- [51] Cai-Nicolas Ziegler, Sean M. McNee, Joseph a. Konstan, and Georg Lausen. Improving recommendation lists through topic diversification. *Proceedings of the 14th international conference on World Wide Web - WWW '05*, page 22, 2005.

Appendix A

Glossary

In this appendix we give an overview of frequently used terms and abbreviations.

Cold-start problem: ... A cold-start problem arises when a new user, item or domain is added to a recommender system, creating a situation in which not enough information about users' preferences for items is available to come up with good recommendations

Data: ... A digital form of an item

Domain: ... A domain is the name of a group of items that share a certain characteristic. In this thesis this characteristic is video type and examples of domains are news items and documentaries

Item: ... Something that can be recommended to a user, e.g. the movie "The Matrix" or the book "Harry Potter and the Goblet of Fire"

Item profile: ... A set of relevant features characterizing an item, used in a recommender system to determine the recommendations. Which features are relevant is determined manually

Meta data: ... Meta data describes the properties of data

Recommender system: ... A system that produces personalized recommendations as output or has the effect of guiding the user in a personalized way to interesting or useful items in a large space of possible options

User: ... An individual person or a group of persons that interacts with a system

User profile: ... A set of relevant features characterizing a user, used in a recommender system to determine the recommendations. Which features are relevant is determined manually

Appendix B

Metadata Open Images videos

Table B.1: Metadata of videos in Open Images format
(<http://openbeelden.nl/api/>)

Element	Explanation	Remarks
oi:title	Title of the item	-
oi:alternative	Subtitle of the item (optional)	-
oi:creator	The creator/producer of the item.	In the case of a person name the format "surname, name" is used, with optional brackets with a role included. For example: "Doe, John (producer)"
oi:subject	Words that describe the item, usually from a closed vocabulary (thesaurus).	This includes person names of people that are present in the media item. These follow the same format as above. Multiple keywords are possible.
oi:description	An introductory description of the item.	-
oi:abstract	A detailed description of the item.	-
oi:publisher	The uploader of the item to Open Images.	For this field are two values are present, the user name and a URL to the profile of the user on Open Images.
oi:contributor	Persons/entities that have contributed to the creation of the item.	In the case of person names, the same format as mentioned above is used. Multiple values are possible.
oi:date	The original publication date of the item.	By default, this is the moment of uploading to Open Images, users can adjust this manually (if necessary).

Continued on next page

Table B.1 – Continued from previous page

Element	Explanation	Remarks
oi:type	The media type of the item.	Items on Open Images are of the types video, audio or still image and are indicated by: http://dublincore.org/documents/dcmi-type-vocabulary/
oi:extent	The length of the item.	The duration is indicated by: http://en.wikipedia.org/wiki/ISO_8601\#Durations
oi:medium	The various formats in which the item is available Open Images.	There are always multiple formats of an item present.
oi:identifier	The catalog number of the item (if derived from an existing collection).	-
oi:source	A reference to the original carrier/source of the items (if any).	-
oi:language	The language the items themselves (not the description).	This value is indicated by the ISO 639-1 standard.
oi:references	A statement about the sources from which the item is a derivative (if any).	-
oi:spatial	The geographic location(s) of the item.	Usually this is a written place name. Multiple values are possible.
oi:attributionName	The name of one or more makers, in the case of reuse this information needs to be mentioned for proper attribution.	In the case of person names, the same format as mentioned above is used. Multiple values are possible.
oi:attributionURL	The location of the original item that, in the case of reuse of the item, should be referenced.	The value of this field is expressed in the form of a URL that refers to the item on Open Images. For example: "http://www.openimages.eu/media/23173"

Continued on next page

Table B.1 – *Continued from previous page*

Element	Explanation	Remarks
oi:license	The license conditions under which the item has been made available.	All items on Open Images are available under a Creative Commons license or are in the public domain. The value of this field is expressed in the form of a URL. For example: "http://creativecommons.org/licenses/by-sa/3.0/nl/deed.en"

Appendix C

Metadata YouTube videos

```
{
  "kind": "youtube#video",
  "etag": etag,
  "id": string,
  "snippet": {
    "publishedAt": datetime,
    "channelId": string,
    "title": string,
    "description": string,
    "thumbnails": {
      (key): {
        "url": string,
        "width": unsigned integer,
        "height": unsigned integer
      }
    },
    "channelTitle": string,
    "tags": [
      string
    ],
    "categoryId": string,
    "liveBroadcastContent": string
  },
  "contentDetails": {
    "duration": string,
    "dimension": string,
    "definition": string,
    "caption": string,
    "licensedContent": boolean,
    "regionRestriction": {
      "allowed": [
        string
      ],
    },
  },
}
```

```
    "blocked": [
      string
    ]
  },
  "contentRating": {
    "acbRating": string,
    "agcomRating": string,
    "anatelRating": string,
    "bbfcRating": string,
    "bfvcRating": string,
    "bmukkRating": string,
    "catvRating": string,
    "catvfrRating": string,
    "cbfcRating": string,
    "cccRating": string,
    "cceRating": string,
    "chfilmRating": string,
    "chvrsRating": string,
    "cicfRating": string,
    "cnaRating": string,
    "csaRating": string,
    "cscfRating": string,
    "czfilmRating": string,
    "djctqRating": string,
    "eefilmRating": string,
    "egfilmRating": string,
    "eirinRating": string,
    "fcbmRating": string,
    "fcoRating": string,
    "fmocRating": string,
    "fpbRating": string,
    "fskRating": string,
    "grfilmRating": string,
    "icaaRating": string,
    "ifcoRating": string,
    "ilfilmRating": string,
    "incaaRating": string,
    "kfcbrating": string,
    "kijkwijzerRating": string,
    "k mrbRating": string,
    "lsfRating": string,
    "mccaaRating": string,
    "mccypRating": string,
    "mdaRating": string,
    "medietilsynetRating": string,
    "mekuRating": string,
    "mibacRating": string,
```

```
    "mocRating": string,
    "moctwRating": string,
    "mpaaRating": string,
    "mtrcbRating": string,
    "nbcRating": string,
    "nbcplRating": string,
    "nfrcbRating": string,
    "nfvcRating": string,
    "nkclvRating": string,
    "oflcRating": string,
    "pefilmRating": string,
    "rcnofRating": string,
    "resorteviolenciaRating": string,
    "rtcRating": string,
    "rteRating": string,
    "russiaRating": string,
    "skfilmRating": string,
    "smaisRating": string,
    "smsaRating": string,
    "tvpgrating": string,
    "ytRating": string
  }
},
"status": {
  "uploadStatus": string,
  "failureReason": string,
  "rejectionReason": string,
  "privacyStatus": string,
  "publishAt": datetime,
  "license": string,
  "embeddable": boolean,
  "publicStatsViewable": boolean
},
"statistics": {
  "viewCount": unsigned long,
  "likeCount": unsigned long,
  "dislikeCount": unsigned long,
  "favoriteCount": unsigned long,
  "commentCount": unsigned long
},
"player": {
  "embedHtml": string
},
"topicDetails": {
  "topicIds": [
    string
  ],

```

```
    "relevantTopicIds": [
      string
    ]
  },
  "recordingDetails": {
    "locationDescription": string,
    "location": {
      "latitude": double,
      "longitude": double,
      "altitude": double
    },
    "recordingDate": datetime
  },
  "fileDetails": {
    "fileName": string,
    "fileSize": unsigned long,
    "fileType": string,
    "container": string,
    "videoStreams": [
      {
        "widthPixels": unsigned integer,
        "heightPixels": unsigned integer,
        "frameRateFps": double,
        "aspectRatio": double,
        "codec": string,
        "bitrateBps": unsigned long,
        "rotation": string,
        "vendor": string
      }
    ],
    "audioStreams": [
      {
        "channelCount": unsigned integer,
        "codec": string,
        "bitrateBps": unsigned long,
        "vendor": string
      }
    ],
    "durationMs": unsigned long,
    "bitrateBps": unsigned long,
    "recordingLocation": {
      "latitude": double,
      "longitude": double,
      "altitude": double
    },
    "creationTime": string
  },
},
```



```
"processingDetails": {
  "processingStatus": string,
  "processingProgress": {
    "partsTotal": unsigned long,
    "partsProcessed": unsigned long,
    "timeLeftMs": unsigned long
  },
  "processingFailureReason": string,
  "fileDetailsAvailability": string,
  "processingIssuesAvailability": string,
  "tagSuggestionsAvailability": string,
  "editorSuggestionsAvailability": string,
  "thumbnailsAvailability": string
},
"suggestions": {
  "processingErrors": [
    string
  ],
  "processingWarnings": [
    string
  ],
  "processingHints": [
    string
  ],
  "tagSuggestions": [
    {
      "tag": string,
      "categoryRestricts": [
        string
      ]
    }
  ],
  "editorSuggestions": [
    string
  ]
},
"liveStreamingDetails": {
  "actualStartTime": datetime,
  "actualEndTime": datetime,
  "scheduledStartTime": datetime,
  "scheduledEndTime": datetime,
  "concurrentViewers": unsigned long
}
}
```


Appendix D

Results YouTube Experiment

Singular	Total results	Plural	Total results	#Overlap in top-5
Paasei	663	Paaseieren	685	2
Vliegtuig	737	Vliegtuigen	721	0
Paard	751	Paarden	751	0
Hond	725	Honden	744	0
Winkel	737	Winkels	731	0
Color	776	Colors	533	1
Phone	749	Phones	775	1
Book	766	Books	740	0
Airplane	726	Airplanes	757	2
Bike	755	Bikes	735	3

Table D.1: Does it make sense to use the singular or plural form of a word?

With stopword	Without stopword	#Overlap in top-5
Een kerstboom versieren	Kerstboom versieren	3
Amsterdam de Dam	Amsterdam Dam	4
De Februaristaking	Februaristaking	4
Een winkelstraat in Rotterdam	Winkelstraat Rotterdam	1
Een winkelstraat in Rotterdam	Winkelstraat in Rotterdam	1
Dog in the snow	Dog in snow	5
Dog in the snow	Dog snow	3
An airplane crash in Dubai	Airplane crash Dubai	4
A cup of coffee	Cup coffee	1
A phone in a microwave	Phone microwave	5

Table D.2: Do stop words influence the results?

Word	Stem	#Overlap in top-5
Lichamelijk	Licham	0
Ophaal	Ophal	0
Ophouden	Ophoud	0
Grijnzen	Grijnz	0
Gingen	Ging	0
Catlike	Cat	0
Fisher	Fish	0
Arguing	Argu	0
Stemming	Stem	0
Speaking	Speak	0

Table D.3: Does YouTube make use of stemming?

Word	Synonym	#Overlap in top-5
Chocolade	Chocola	0
Kerst	Kerstmis	0
Vliegtuig	Vliegmaschine	0
Winkel	Boetiek	0
Paard	Knol	0
Book	Novel	0
Airplane	Aircraft	0
Airplane	Plane	0
Airplane	Jet	0
Dog	Hound	0

Table D.4: Does YouTube make use of synonym expansion?

Word 1	Word 2	#Overlap in top-5
Rennen	Rende	0
Lezen boek	Las boek	0
Duiken water	Dook water	0
Imiteren	Imiteerde	1
Imiteren	Imitatie	1
Imitation	Imitate	1
Dive	Diving	1
Dive water	Diving water	4
Ring telephone	Rang telephone	0
Run technique	Running technique	3

Table D.5: Does the form of a word matter?

Sentence 1	Sentence 2	#Overlap in top-5
Kerstboom chocolade pasen	Pasen chocolade kerstboom	2
Airshow seppe	Seppe airshow	4
Februaristaking Amsterdam	Amsterdam februaristaking	4
Chocolade paasei kerst	Kerst chocolade paasei	5
Chocolade paasei kerst	Kerst paasei chocolade	2
Dog snow America	America snow dog	2
Dog snow America	Snow America dog	4
Airplane crash Dubai	Dubai crash airplane	4
Cup coffee	Coffee cup	0
Phone microwave	Microwave phone	4

Table D.6: Does the order of words matter?

Appendix E

Results Offline Experiment

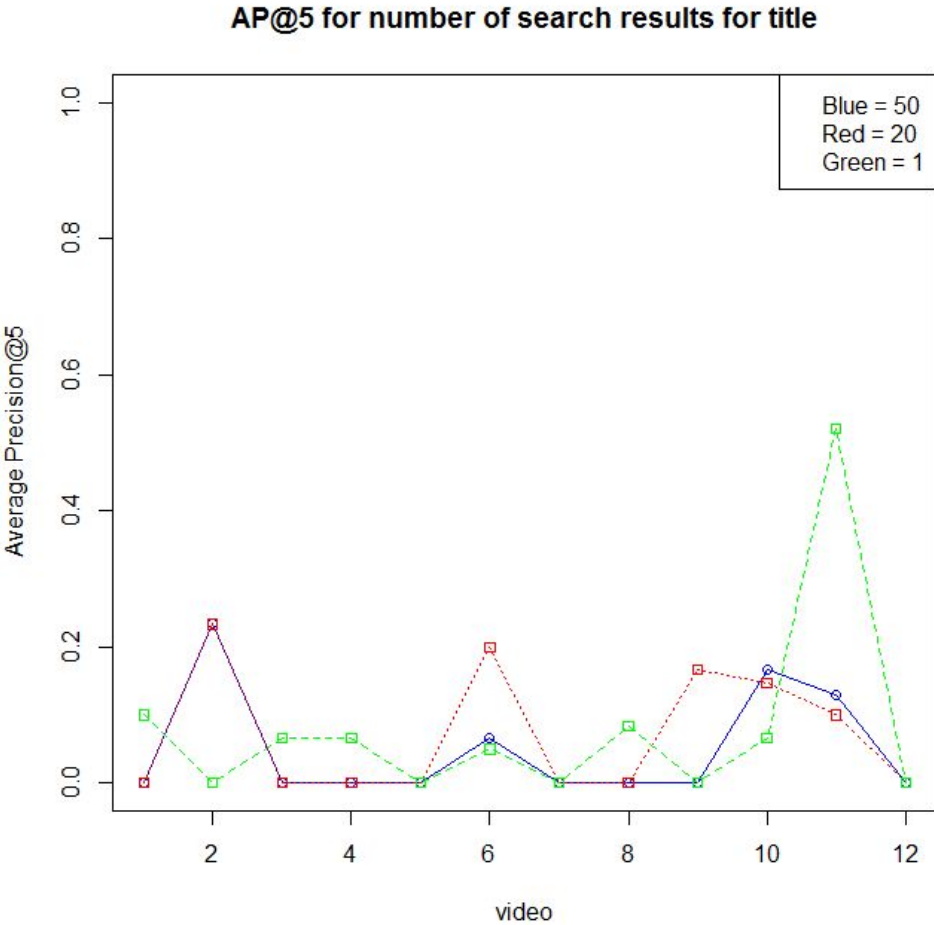


Figure E.1: Average Precision@5 for the number of search results for the strategy Title

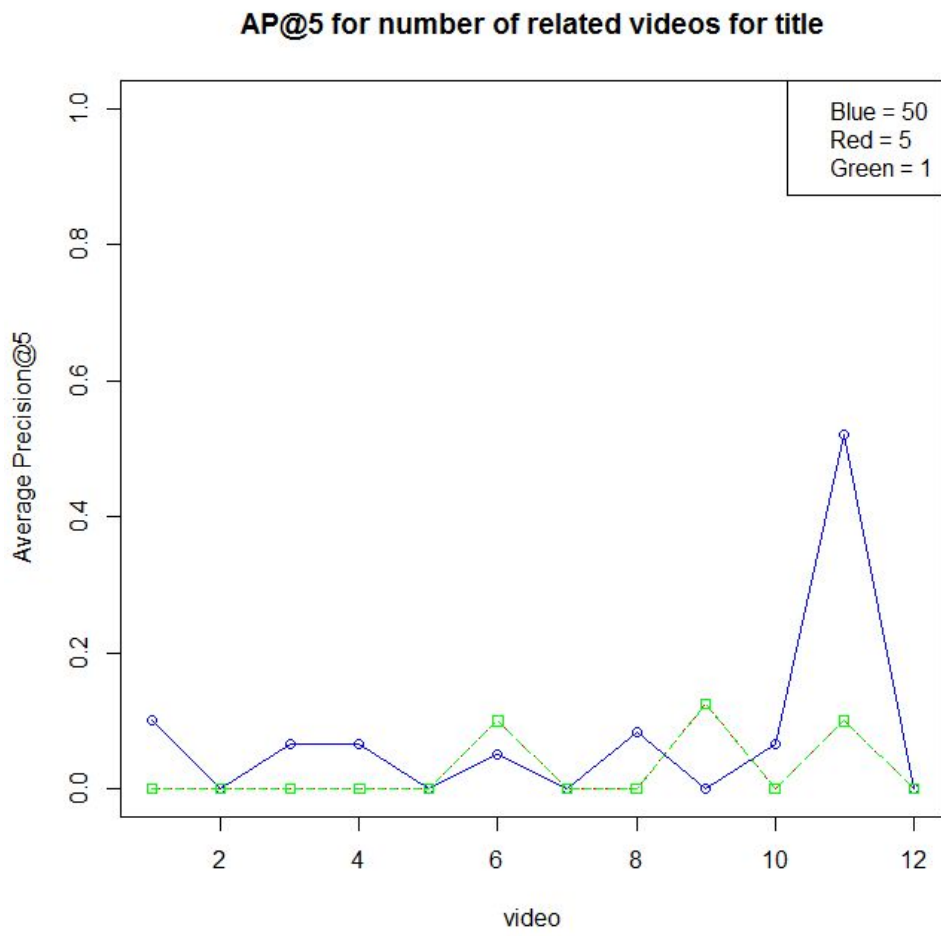


Figure E.2: Average Precision@5 for the number of related videos for the strategy Title

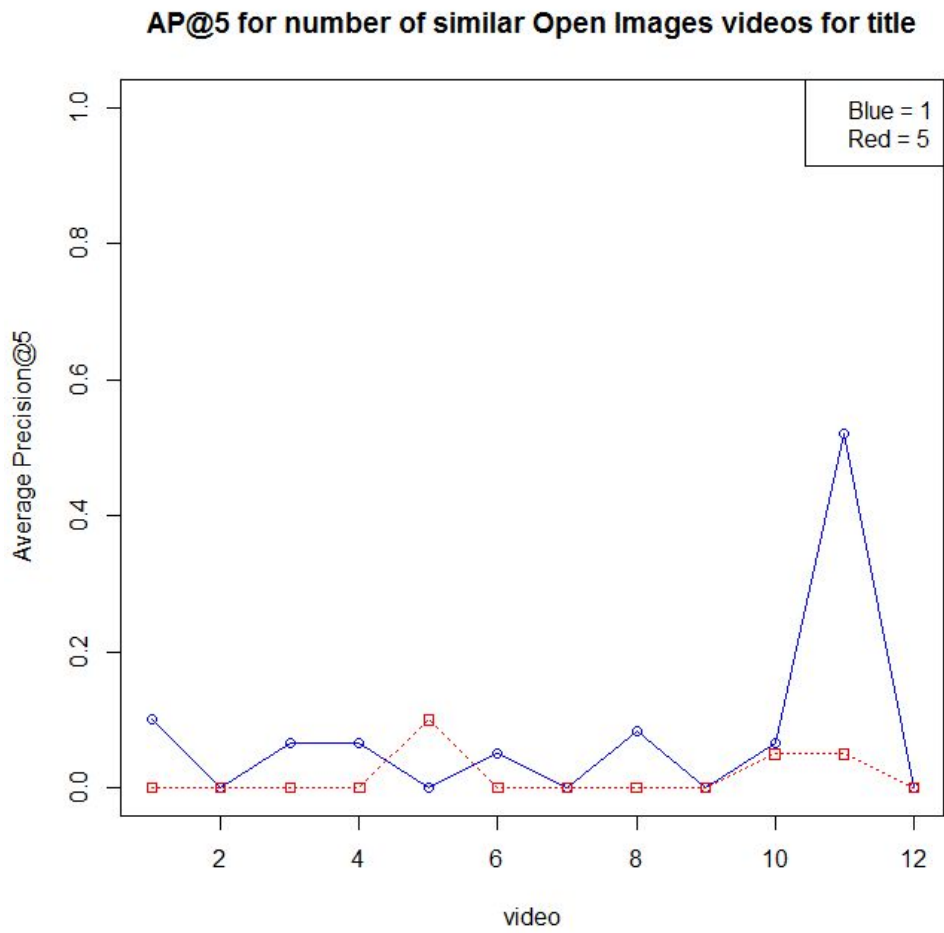


Figure E.3: Average Precision@5 for the number of similar Open Images videos for a given YouTube video for the strategy Title

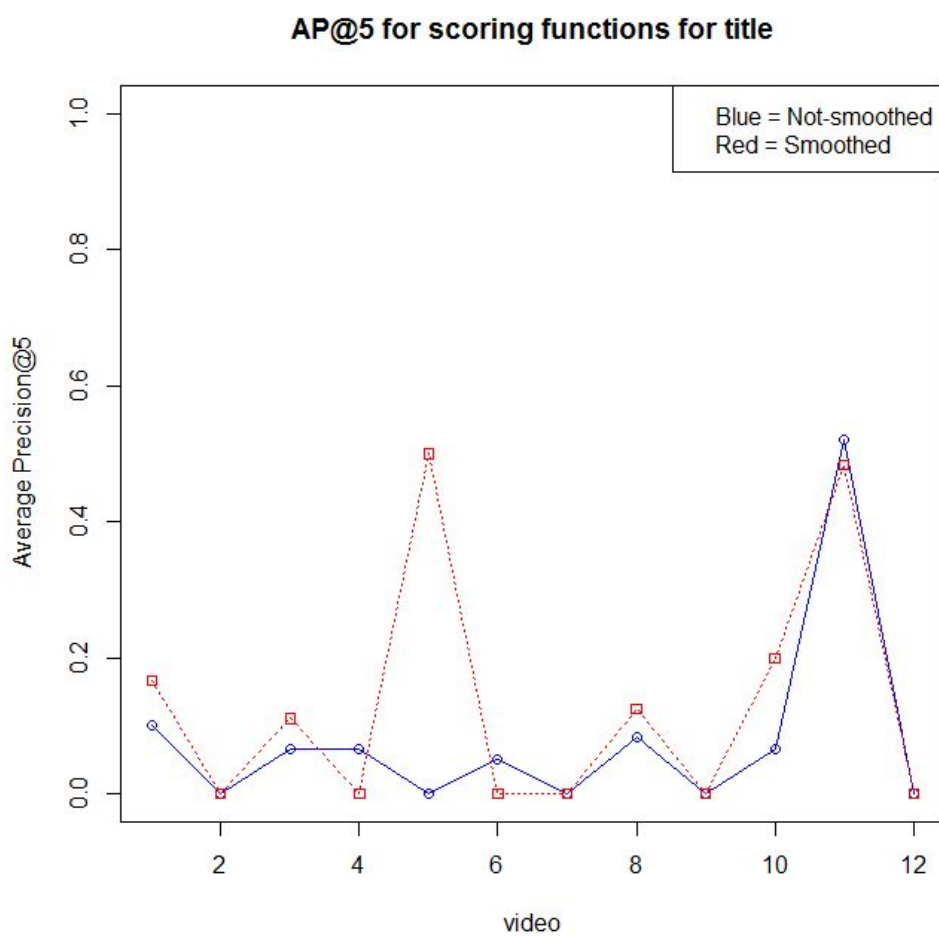


Figure E.4: Average Precision@5 for the different scoring functions for the strategy Title

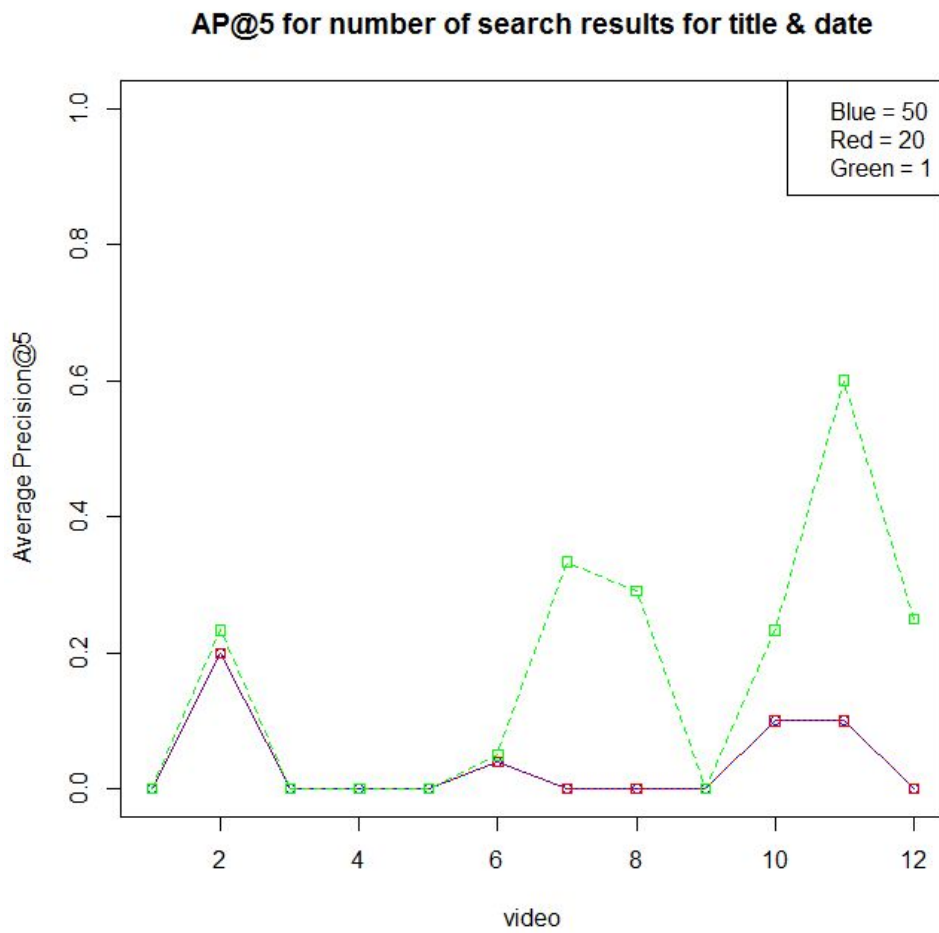


Figure E.5: Average Precision@5 for the number of search results for the strategy Title & Date

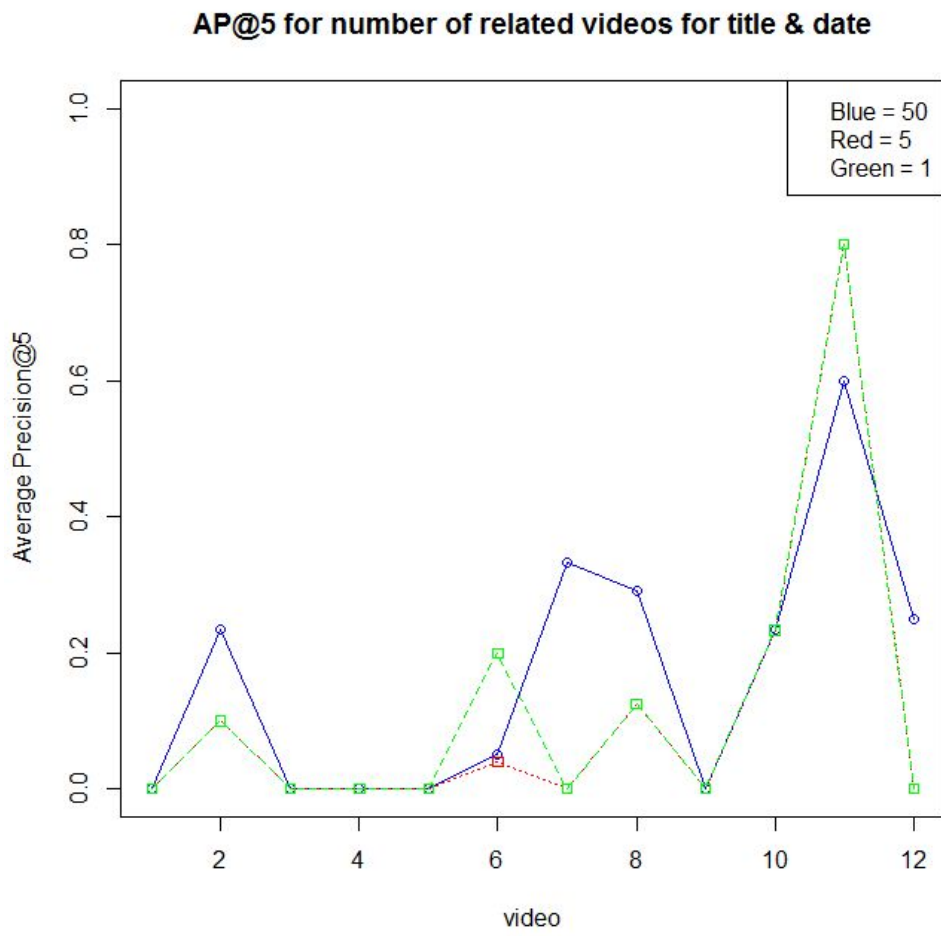


Figure E.6: Average Precision@5 for the number of related videos for the strategy Title & Date

AP@5 for number of similar Open Images videos for title & date

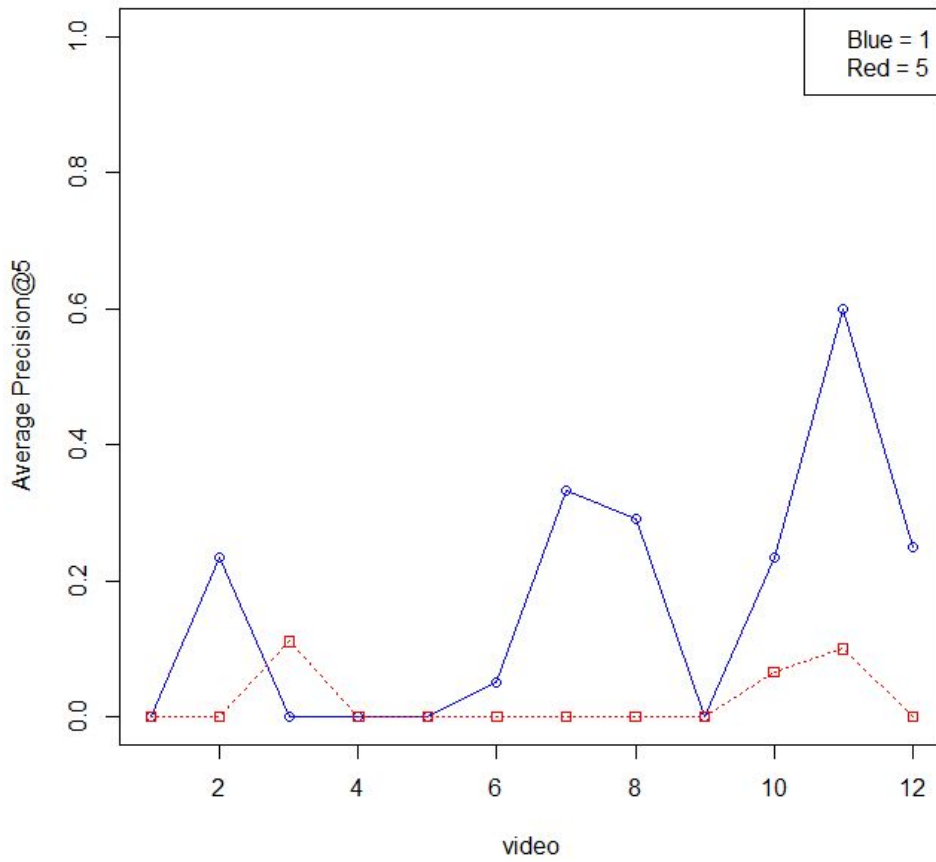


Figure E.7: Average Precision@5 for the number of similar Open Images videos for a given YouTube video for the strategy Title & Date

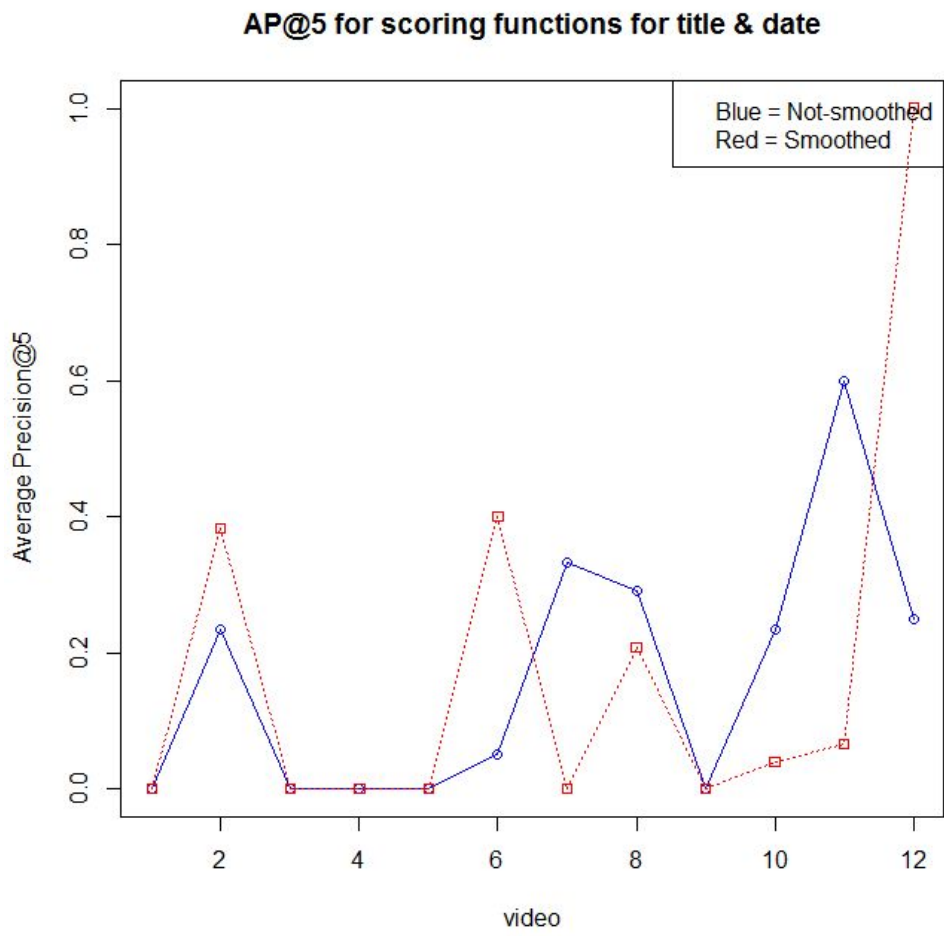


Figure E.8: Average Precision@5 for the different scoring functions for the strategy Title & Date

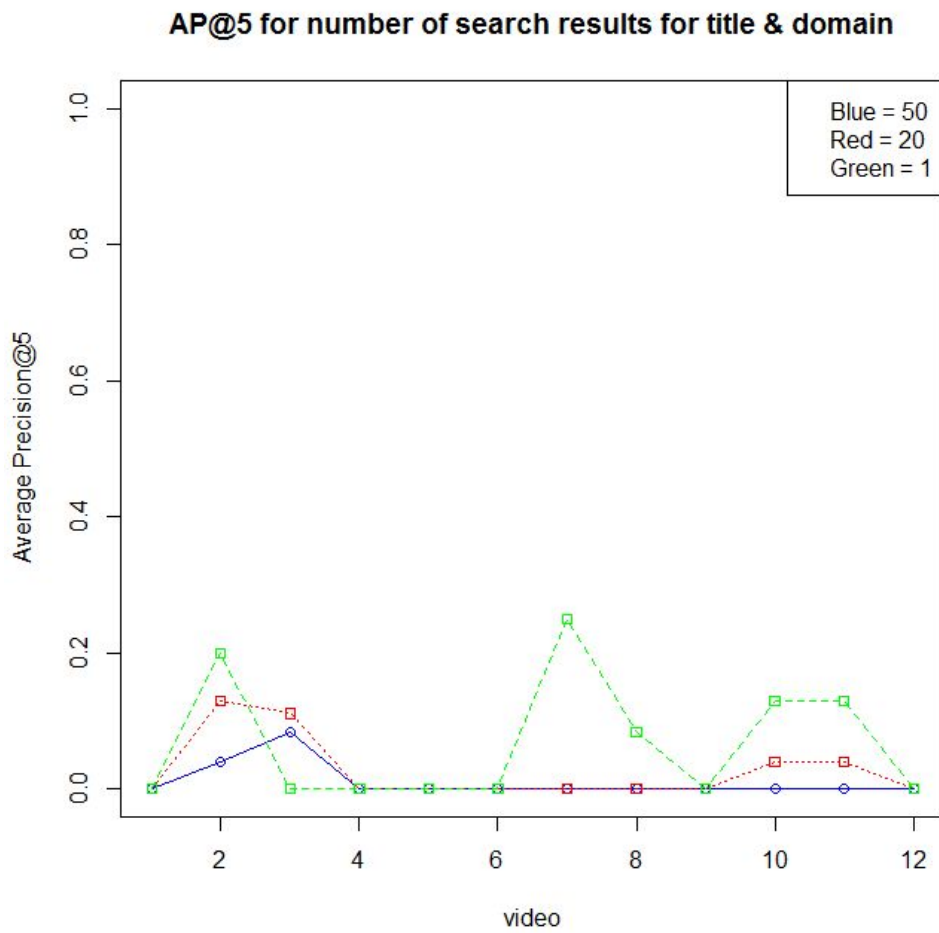


Figure E.9: Average Precision@5 for the number of search results for the strategy Title & Domain

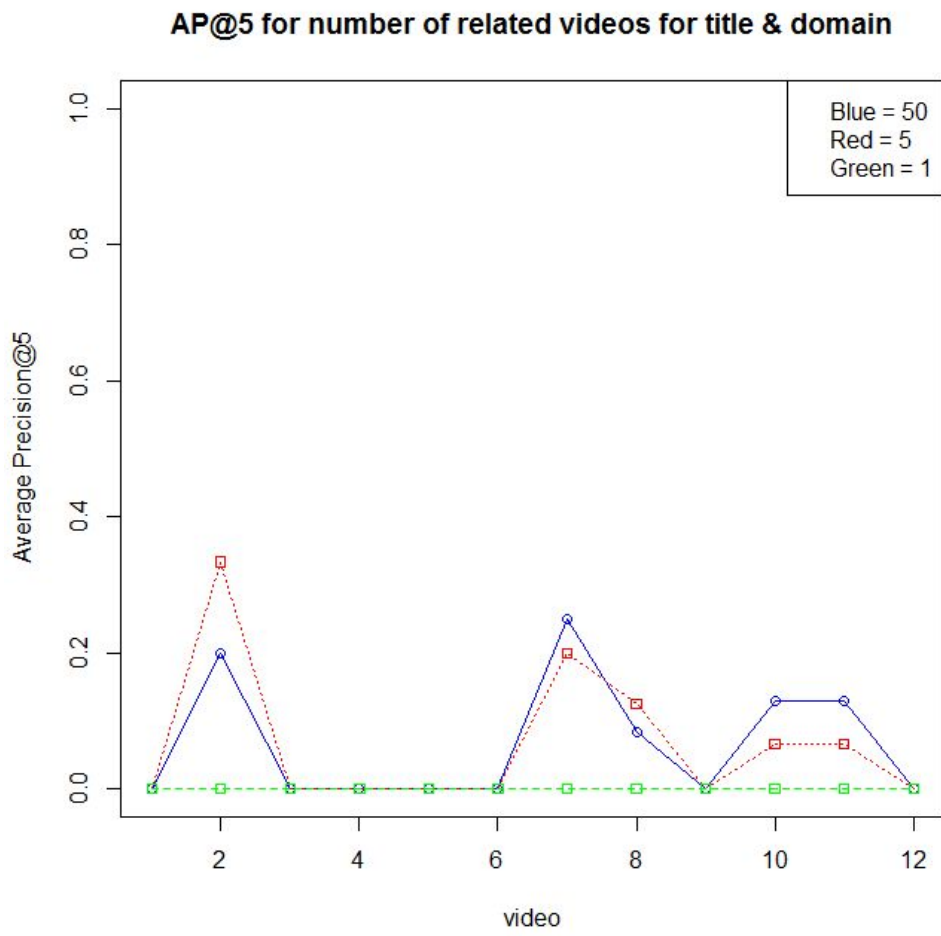


Figure E.10: Average Precision@5 for the number of related videos for the strategy Title & Domain

AP@5 for number of similar Open Images videos for title & domain

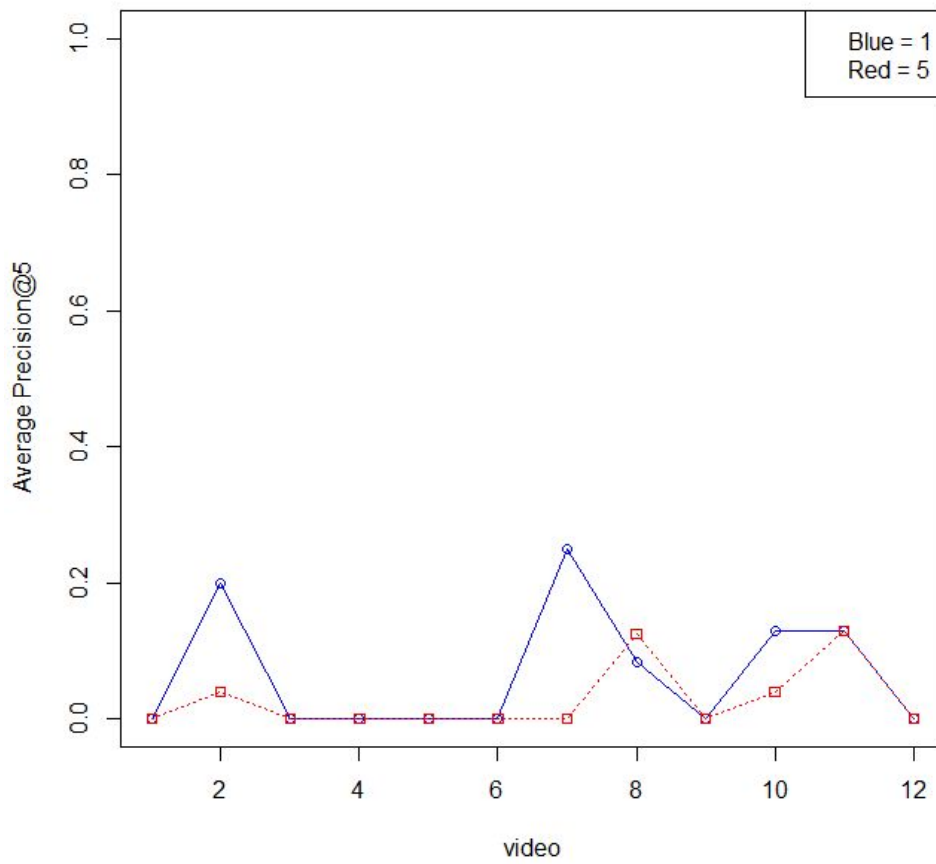


Figure E.11: Average Precision@5 for the number of similar Open Images videos for a given YouTube video for the strategy Title & Domain

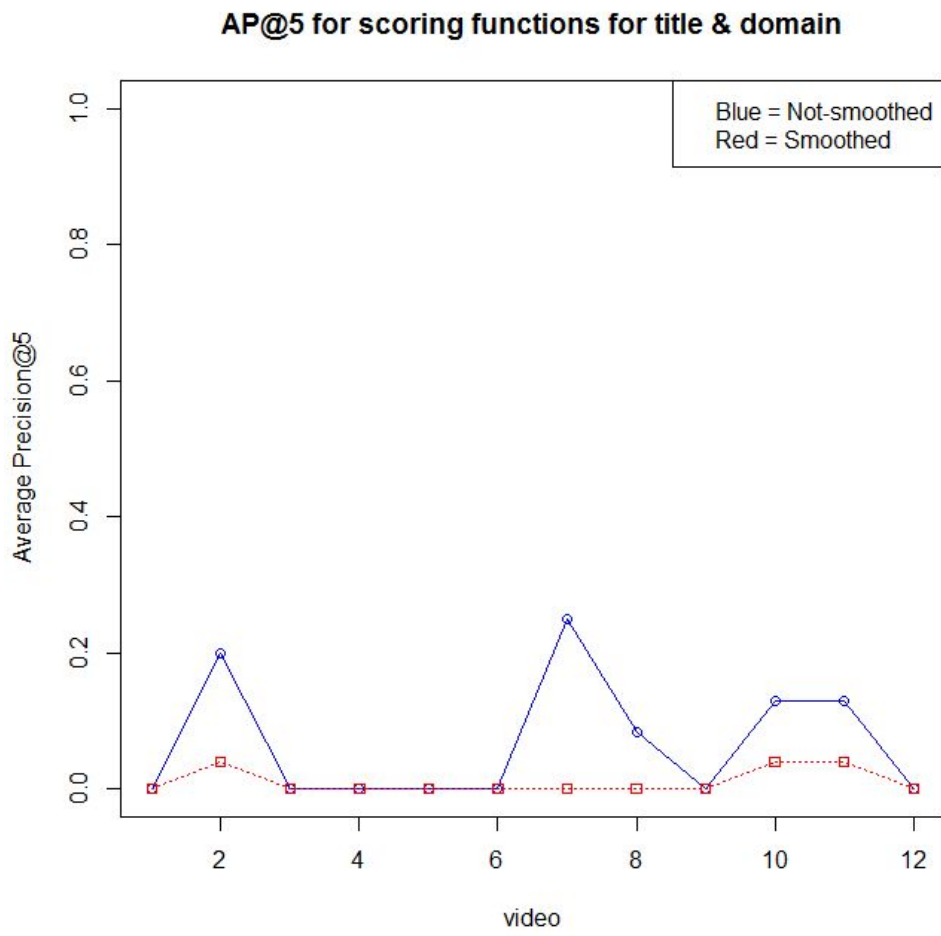


Figure E.12: Average Precision@5 for the different scoring functions for the strategy Title & Domain

Appendix F

Results majority voting User Study

See next page...

First	Second	N1	N2	N3	N4	N5	D1	D2	D3	D4	D5	E1	E2	E3	E4	E5	#Videos 1	#Videos 2	# -
1	2	2	2	2	-	2	1	2	1	1	1	1	-	-	1	-	6	5	4
1	3	-	-	3	1	3	1	3	3	-	1	1	3	-	1	3	5	6	4
1	4	1	4	4	-	1	1	1	1	1	1	1	4	1	4	4	9	5	1
1	5	1	-	5	5	5	5	-	5	1	-	5	5	1	5	5	3	9	3
2	3	-	-	2	3	2	2	3	3	2	3	2	3	3	2	3	6	7	2
2	4	2	4	4	4	2	2	2	2	-	-	2	4	2	-	-	7	4	4
2	5	2	5	-	5	2	5	2	5	2	5	5	5	2	5	5	5	9	1
3	4	3	-	4	4	3	4	3	3	-	3	3	4	3	4	3	8	5	2
3	5	3	5	5	3	5	5	3	5	3	5	5	5	3	5	5	5	10	0
4	5	4	5	4	5	5	5	-	5	-	5	5	-	5	5	5	2	10	3

Table F.1: Best strategy using majority voting per pair per video. 1 = Title, 2 = Title & Date, 3 = Title & Domain, 4 = Random, 5 = Tags. N=News item, D=Documentary/Report, E=Event coverage

Appendix G

Comments User Study

- **Comments on video news item 1:**
"What is the comparison between Nieuw Guinea and Heidekoningin ter schapenmarkt in Ede?"
- **Comments on video news item 4:**
"For some movies it is hard to determine whether it is a good recommendation. There is no way of viewing the video that are recommended, not even a snippet."
- **Comments on video news item 5:**
"Didn't find any of the recommendation really matched the given video. So I used what came as close as possible."
- **Comments on video documentary/report 2:**
 1. "Not so much different lists."
 2. "Some recommendation lists had no image and title for 4/5 videos"
 3. "It's pretty hard to decide on these lists in some cases; while I can sometimes see how certain videos could be related, there was not always a clear advantage to either list"
- **Comments on video documentary/report 3:**
"Sometimes it is hard to compare, also sometimes left and right are equal in comparison. Perhaps let the user select the recommendations they think are appropriate?"
- **Comments on video documentary/report 4:**
"I think only one video was a good recommendation for the documentary"
- **Comments on video event coverage 5:**
 1. "Strange that in the comparison there are continue the same videos. And a lot of the recommendations do have nothing to do with the video, that makes it really tough for some comparisons.."
 2. "Sets are almost always too similar to make a good decision"

Appendix H

Recommendations for video documentary/report 5

See next page...

Title (20)	Title & Date (5)	Title & Domain (10)	Random (9)	Tags (16)
Grand prix waterski-race op het Veluwemeer Motortereinwedstrijden	Vleugel van nieuw stad-huis gereed Record melkgift van koe	Demonstratie met pantserwagens Waterloper verbaasd schaapvaart, loopt op friese meren en kanalen	Estafette in Vondelpark Concours d'elegance	Wedstrijd van vliegtuig-modellen Nationale ruilbeurs op de dag der verzamelaars
Demonstratie met pantserwagens	Ontheemden uit Midden-Europa vinden asiel in ons land	Burgers (des)organiseren het verkeer in Eindhoven	Burgers (des)organiseren het verkeer in Eindhoven	Raketten, gevaarlijk speelgoed
Kampeertentoonstelling	Padvinderswaterkamp	Jongelui leren banket-bakkersvak	Vakantietijd	-
De mooie, warme zomer	Grand prix waterski-race op het Veluwemeer	De mooie, warme zomer	Kampeertrein voor de Amsterdamse jeugd	-

Table H.1: Recommendations for Documentary/Report 5. The number of votes for each strategy is shown between the brackets