NoTube

Networks and ontologies for the transformation and unification of broadcasting and the Internet

FP7—231761

D3.3 Recommender service

Coordinator: Davide Palmisano (Pronetics)

With contributions from: Libby Miller, Vicky Buser (BBC), Michele Minno, Davide Palmisano (Pronetics), Dan Brickley, Balthasar Schopman, Lora Aroyo (VU)

Quality Assessor: Guus Schreiber (VU)

Quality Controller: Davide Palmisano (Pronetics)

<table>
<thead>
<tr>
<th>Document Identifier</th>
<th>NoTube/2011/D3.3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Version</td>
<td>2</td>
</tr>
<tr>
<td>Date</td>
<td>14/11/2011</td>
</tr>
<tr>
<td>State</td>
<td>Final</td>
</tr>
<tr>
<td>Distribution</td>
<td>PU</td>
</tr>
</tbody>
</table>
EXECUTIVE SUMMARY

This document is a detailed description of the investigations and experimentation underlying the Recommendation Service, within the scope of Workpackage 3. This document focuses on the design considerations and experimental lessons learned for the deployment of the recommendation solutions. In section 2 we will introduce the approach to hybrid TV content recommendations. In section 3 we concentrate on the semantics-based strategies, while in section 4 we turn to the statistics-based strategies. We describe features and characteristics for the strategies and their implications. In section 5 we pay attention to the element of explanations and user feedback, as an element of the approach to recommend relevant TV content. Section 6 describes the experiences from the deployment in the NoTube use cases, while in sections 7 and 8 we consider the implementations and datasets used respectively. After considering the privacy aspect in Section 9, we conclude this document with the view on the progress and future work for the Recommendation Service.
This document is a detailed description of the investigations and experimentation underlying the Recommendation Service, within the scope of Workpackage 3. This document focuses on the design considerations and experimental lessons learned for the deployment of the recommendation solutions with respect to achieving the main goal of the NoTube project - to help people find and choose what TV content to watch in a continuously growing amount of TV content and channels. We have experimented with novel semantics-based approaches for recommending TV programs. The core goal of such approaches is to allow for finding new connections between TV programs (e.g. connections between people involved in TV programs). Central to those approaches is using Linked Data as background knowledge in order to make such connections more specific and meaningful. Further, our goal is to integrate the semantics-based approaches in a complementary fashion to optimize the finding of recommendation results for any user and in any context. We have shown how to summarize Linked Data sets by treating them as sets of connected knowledge patterns, in order to identify their core knowledge components. Next to these semantic pattern-based approaches, the recommendation service also uses standard statistical techniques for collaborative filtering based on rating data. In reality, such TV data is very sparse, and expanding TV data like this can bring more programs from the long tail to the user's attention. Another aspect of the recommendation is matching more directly the concepts in the user profile, possible distilled from the user's activity data, with concepts related to programs, for example as basis for news-related recommendations. These points were mainly covered in Sections 2, 3, 4 and 5. Section 6 describes the experiences from the deployment in the NoTube use cases, while in sections 7 and 8 we consider the implementations and datasets used respectively. After considering the privacy aspect in Section 9, we conclude this document with the view on the progress and future work for the Recommendation Service.
# PROJECT CONSORTIUM INFORMATION

<table>
<thead>
<tr>
<th>Participants</th>
<th>Contact</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vrije Universiteit Amsterdam</td>
<td>Guus Schreiber</td>
</tr>
<tr>
<td></td>
<td>Phone: +31 20 598 7739/7718</td>
</tr>
<tr>
<td></td>
<td>Email: <a href="mailto:schreiber@cs.vu.nl">schreiber@cs.vu.nl</a></td>
</tr>
<tr>
<td>British Broadcasting Corporation</td>
<td>Libby Miller</td>
</tr>
<tr>
<td></td>
<td>Phone: +44 787 65 65 561</td>
</tr>
<tr>
<td></td>
<td>Email: <a href="mailto:Libby.Miller@bbc.co.uk">Libby.Miller@bbc.co.uk</a></td>
</tr>
<tr>
<td>Pro-netics</td>
<td>Marco Bruni</td>
</tr>
<tr>
<td></td>
<td>Phone: +39 06 45472503</td>
</tr>
<tr>
<td></td>
<td>Email: <a href="mailto:marco.bruni@pro-netics.com">marco.bruni@pro-netics.com</a></td>
</tr>
<tr>
<td>Engin Medya Hizmetleri A.S.</td>
<td>Ron van der Heiden</td>
</tr>
<tr>
<td></td>
<td>Phone: +31 6 2003 2006</td>
</tr>
<tr>
<td></td>
<td>Email: <a href="mailto:ron@engin.tv">ron@engin.tv</a></td>
</tr>
<tr>
<td>Institut fuer Rundfunktechnik GmbH</td>
<td>Christoph Dosch</td>
</tr>
<tr>
<td></td>
<td>Phone: +49 89 32399 349</td>
</tr>
<tr>
<td></td>
<td>Email: <a href="mailto:dosch@irt.de">dosch@irt.de</a></td>
</tr>
<tr>
<td>Ontotext AD</td>
<td>Atanas Kiryakov</td>
</tr>
<tr>
<td></td>
<td>Phone: +35 928 091 565</td>
</tr>
<tr>
<td></td>
<td>Email: <a href="mailto:naso@sirma.bg">naso@sirma.bg</a></td>
</tr>
<tr>
<td>Open University</td>
<td>Stefan Dietze</td>
</tr>
<tr>
<td></td>
<td>Phone: +44 1908 858 217</td>
</tr>
<tr>
<td></td>
<td>Email: <a href="mailto:s.dietze@open.ac.uk">s.dietze@open.ac.uk</a></td>
</tr>
<tr>
<td>RAI Radiotelevisione Italiana SPA</td>
<td>Alberto Morello</td>
</tr>
<tr>
<td></td>
<td>Phone: +39 011 810 31 07</td>
</tr>
<tr>
<td></td>
<td>Email: <a href="mailto:a.morello@rai.it">a.morello@rai.it</a></td>
</tr>
<tr>
<td>Semantic Technology Institute International</td>
<td>Lyndon Nixon</td>
</tr>
<tr>
<td></td>
<td>Phone: +43 1 23 64 002</td>
</tr>
<tr>
<td></td>
<td>Email: <a href="mailto:lyndon.nixon@sti2.org">lyndon.nixon@sti2.org</a></td>
</tr>
<tr>
<td>Stoneroos B.V.</td>
<td>Annelies Kaptein</td>
</tr>
<tr>
<td></td>
<td>Phone: +31 35 628 47 22</td>
</tr>
<tr>
<td></td>
<td>Email: annelies.kaptein@stoneroos</td>
</tr>
<tr>
<td>Thomson Video Networks</td>
<td>Raoul Monnier</td>
</tr>
<tr>
<td></td>
<td>Phone: +33 2 99 27 30 57</td>
</tr>
<tr>
<td></td>
<td>Email: <a href="mailto:raoul.monnier@thomson-networks.com">raoul.monnier@thomson-networks.com</a></td>
</tr>
<tr>
<td>Polymedia, SpA</td>
<td>Tullio Pirovano</td>
</tr>
<tr>
<td></td>
<td>Phone: +39 02 257711</td>
</tr>
<tr>
<td></td>
<td>Email: <a href="mailto:tullio.pirovano@polymedia.it">tullio.pirovano@polymedia.it</a></td>
</tr>
<tr>
<td>KT Corporation</td>
<td>Myoung-Wan Koo</td>
</tr>
<tr>
<td></td>
<td>Phone: +82 2 526 6347</td>
</tr>
<tr>
<td></td>
<td>Email: <a href="mailto:mskim@kt.co.kr">mskim@kt.co.kr</a></td>
</tr>
</tbody>
</table>
# TABLE OF CONTENTS

List of Tables .................................................................................................................................................. 6

List of Figures ................................................................................................................................................ 7

1. Introduction ................................................................................................................................................... 8

2. Hybrid Approach to TV Content Recommendations .................................................................................. 9

3. Semantics-based Recommendation Strategies .......................................................................................... 10
   3.1. Background ............................................................................................................................................. 10
   3.2. Property-based Recommendations ...................................................................................................... 10
   3.3. Linked Data patterns ............................................................................................................................. 13

4. Statistics-based Recommendation Strategies .......................................................................................... 21

5. Explanations and User Feedback ............................................................................................................. 24
   5.1. Rationale for Explanations of Recommendations ............................................................................... 25
   5.2. Experiment with the Notion of Serendipity ....................................................................................... 26
   5.3. Experiment with Explanations and Related Programs ....................................................................... 26

6. Deployment in NoTube Use Cases ........................................................................................................... 29
   6.1. Semantic News Recommendations (WP7a) .................................................................................... 29
   6.2. Adaptive Ad Inclusion (WP7b) .......................................................................................................... 30
   6.3. Recommendations in KT demonstrator (WP7b) .............................................................................. 30
   6.4. Recommendations in BBC demonstrator (WP7c) ........................................................................... 31

7. Implementations and Recommendation APIs ......................................................................................... 32
   7.1. Implementation of the Property-based recommender ......................................................................... 32
   7.2. Implementation of LOD patterns recommender ................................................................................ 32
   7.3. Implementation of N-Screen recommendation demonstrator .......................................................... 32

8. Datasets ....................................................................................................................................................... 36

9. Privacy ......................................................................................................................................................... 37
   9.1. Data ownership & portability: .............................................................................................................. 37
   9.2. Sharing of data - UI principles: ......................................................................................................... 37
   9.3. Sharing - user control & level of granularity: .................................................................................... 38

10. Conclusion .................................................................................................................................................. 40

11. Works Cited ............................................................................................................................................... 40
List of Tables

**TABLE 1:** Statistics w.r.t. “implicit-participant” pattern: number of triples of all predicates that have a film as subject and a person as object. Variables: ?I is a user interest; ?F is a movie feature that may match a TV show feature.

**TABLE 2:** Influenced-by pattern. Variables are the same as in Table 1 and ?X is an intermediate entity.

**TABLE 3:** The average accuracy and novelty of both patterns and the amount of times evaluators indicated they did not know how to evaluate a recommendation.

**TABLE 4:** Accuracy/novelty distribution table of the «implicit-participants» patterns. The different accuracy evaluations are distributed over the columns. The different novelty evaluations are distributed over the rows.

**TABLE 5:** Accuracy/novelty distribution table of the «influenced-by» patterns. The different accuracy evaluations are distributed over the columns. The different novelty evaluations are distributed over the rows.
List of Figures

FIGURE 1: PROPERTIES-BASED RELATIONS BETWEEN USERS AND MEDIA ITEMS ................................................................. 11
FIGURE 2: LINKED DATA CLOUD SELECTED SOURCES ...................................................................................................... 12
FIGURE 3: PROPERTIES-BASED RECOMMENDATION ALGORITHM .................................................................................... 13
FIGURE 5: INSTANTIATION OF "INFLUENCES-BY" PATTERN. THE SHADED BOXES REPRESENT MEDIA ITEMS. THE STRIPED BOXES INDICATE WHERE THE DATA RESIDES; THE DOTTED NODES INDICATE THE TYPE OF NODES FOR CLARITY REASONS. ............... 16
FIGURE 7: DISTRIBUTION OVER EACH OF THE ANSWERS TO BOTH THE ACCURACY AND THE NOVELTY QUESTIONS OF RECOMMENDATION OF THE TWO TYPES OF THE "INFLUENCED-BY" PATTERN .................................................................................. 19
FIGURE 9: RELATED PROGRAMS IN NOTUBE ARCHIVE BROWSER .......................................................................................... 27
FIGURE 10: RELATED PROGRAMS FOR RAILWAY STATIONS .................................................................................................. 28
FIGURE 11: (A) PROGRAM SIMILARITY (B) DISPLAY OF SUBJECT CATEGORIES ........................................................................ 28
FIGURE 12: COMPARISON OF QUESTIONNAIRE RESULTS BETWEEN CONTROL AND OPTIMAL GROUPS ........................................ 29
FIGURE 13: INTEGRATION WORKFLOW ..................................................................................................................................... 30
FIGURE 14: A CLUSTER OF 'DAYTIME HOME-RELATED FACTUAL' ............................................................................................. 33
FIGURE 15: RELATED ITEMS FOR LIZA DONNELLY VIDEO ..................................................................................................... 34
FIGURE 16: EXAMPLE OF A NOT VERY GOOD SIMILARITY MATCH BUT IN MANY CASES, REASONABLY GOOD .................................................................................................................. 35
FIGURE 17: EXAMPLE OF A GOOD SELECTION OF SIMILAR MATERIAL AND OCCASIONALLY THROWS UP AN INTERESTING SURPRISE ................................................................................ 35
FIGURE 18: UNEXPECTED LINK BETWEEN PROGRAMS ............................................................................................................. 36
1. Introduction

The semantic annotation framework provides general infrastructure for running various annotation services subject of development in the NoTube project. It also implements those tasks that are common for all annotation services e.g. content retrieval and metadata storage. It also provides easy access to external publically available data sources, e.g. DBPedia and LinkedMDB.

The constant expansion trend of Linked Data (LD) is broadening the potential exploitation range of their datasets for improving search through related data. Current research [20,14] and established Web search firms like Google and Powerset show the benefits of using explicit semantics and LD to refine search results. However, using efficiently the explicit knowledge of each dataset can be awkward and ineffective. Datasets typically cover diverse domains, do not follow a unified way of organizing the knowledge, and differ in size, granularity and descriptiveness. To avoid burdensome, dataset-specific querying schemes, the following are required: (1) measures and indicators that provide a landscape view on a dataset; (2) a way to query a dataset even with no prior knowledge of its vocabulary.

We propose an approach to examine LD with these problems in mind. It employs a strategy for inspecting datasets and identifying emerging knowledge patterns (KPs). A key step of this method is the construction of a formal logical architecture, or dataset knowledge architecture, which summarizes the key features and figures of one or more datasets, thus addressing requirement (1). This, in turn, relies on the notions of KPs and type-property paths. We identify the central properties and types, i.e. those able to capture most of the knowledge in a dataset, and extract KPs based on the central types. In other words, we extract the dataset vocabulary and analyze the way the data are used in terms of patterns. We also associate general-purpose measures, such as betweenness and centrality, to the knowledge architecture components of a dataset for performing empirical analysis. These notions and measures will be defined throughout the paper. Using KPs and paths, we can provide prototypical ready-to-use queries for core and concealed knowledge to emerge, thus addressing requirement (2). Although the method applies to datasets whose logical structure is not known a priori, it is meant to analyze LD for serendipitous knowledge. Unlike a mere reverse-engineering exercise, our method discovers new knowledge about datasets, such as their central types and properties and emerging patterns.

The method was partly applied manually and our claims on it are observed empirically, yet it can be generalized and fully automated, as the construction of a dataset architecture and computation of its measures are all derived by directly querying the data using metalevel constructs from RDF and OWL.

In the NoTube project, the main goal is to help people choose what TV content to watch in the light of the enormous growth of TV content available. Therefore, we investigate novel approaches for recommending TV programs. These approaches exploit techniques for finding new connections between TV programs, especially connections between people involved in these programs. For this purpose, we use background knowledge from the Linked Open Data freely available on the web to make such connections more specific and meaningful.

In NoTube's recommendation services we integrate three different approaches with the aim to include suggestions for new content that otherwise would not be found.

The basis for NoTube's recommendation services is the user profile that Beancounter has produced. Beancounter has used the user's activities from the social web and linked them to other data available on the open web, and has thus created a machine-readable user profile representing the user's interests. This interest is represented through weighted concepts. This weighted interest profile helps the recommendation service to generate recommendations.

The recommendation approach has identified rules for following interesting patterns in the cloud of Linked Open Data to connect concepts in a user's profile to concepts of programs to recommend. These interesting patterns reflect interesting relationships between people, such as people that have
creatively influenced each other or that have had joint participations. Such relationships between people are the basis for relationships between programs and thus for recommendations.

Next to these semantic pattern-based approaches, the recommendation service also uses standard statistical techniques for collaborative filtering based on rating data. In reality, such TV data is very sparse, and expanding TV data like this can bring more programs from the long tail to the user's attention.

The third aspect of the recommendation is matching more directly the concepts in the user profile, possible distilled from the user's activity data, with concepts related to programs, for example as basis for news-related recommendations.

2. Hybrid Approach to TV Content Recommendations

In general, TV recommender systems address the issue of information overload, but in doing so typically exploit only a limited set of user activity data and content items available through a single settop box. This user activity data identifies preferences that users express in the closed settop box environments, and as such it is often scarce and outdated. In NoTube we aim at exploiting data from the Social Web, e.g. Facebook and Twitter, which have become a rich source of up-to-date overview of user's interests and preferences in the TV domain. However, most of the Social Web, online streaming or video-on-demand (VOD) websites, still maintain separate, unrelated sets of user preference and activity data, which results in distributed and scattered user profiles all around the Web. That is why, in NoTube we aggregate this scattered Social Web user data it in weighted interests user profiles [See Del 3.4] in order to provide recommendations of TV and VOD content. More details on how the NoTube user profiling service, the Beancounter, aggregates the user activity data from several sources, e.g. music listening activity from last.fm, TV activity from Twitter (using tweet from applications such as Boxee, Trakt and Miso), and general ‘like-button’ data from Facebook, can be found in Del 3.4.

Recent research has shown the Linked Data cloud to be a potentially ideal basis for improving user experience when interacting with Web content across different applications and domains. Using the explicit knowledge of datasets, however, is neither sufficient nor straightforward. Dataset knowledge is often not uniformly organized, thus it is generally unknown how to query for it. To deal with these issues, we propose a dataset analysis approach based on knowledge patterns, and show how the recognition of patterns can support querying datasets even if their vocabularies are previously unknown. This analysis approach based on knowledge patterns brings a hybrid approach to recommendations where semantics-based and statistics-based considerations are combined. This hybrid approach is at the heart of the services in NoTube and in the next sections we consider both aspects, i.e. semantics-based and statistics-based, before turning to the deployment of the hybrid approach in use cases and discussing results from experimenting on three multimedia-related datasets.

Below we give a few examples of the type of recommendations we are targeting. For instance, for a given TV program selected or liked by the user, we would recommend other programs of:

- the same (or related) genre, considering (or excluding) the current user context
- the same (or related) genre, and also with (related) actors/presenters/directors/people who are liked in the user profile (in the current context)
- the same (or related) genre, also with (related) topics/terms who are liked in the user profile (in the current context)
- the same series, which are not watched yet (or which are loved highly without a context)
- the same language group (land) for the current context in the same (or related genre) in the same (or related format, e.g. live or recorded program,

The target is to at the end always achieve some version of a hybrid recommendation strategy, which will consider content features, the user context, and also the features relevant from his 'social circle' friends.
3. Semantics-based Recommendation Strategies

3.1. Background

Semantic Web technology has been used in recommender systems to enhance system interoperability, enable abstraction over data and address the “cold start” problem [8]. However, to the best of our knowledge, Linked Data background knowledge has not been exploited for content-based recommendation strategies. Lops et al. discuss the importance of serendipity in recommender systems, which is strongly related to diversity of recommendations [7]. They explain that according to Gup’s theory content-based recommenders have no inherent method for generating serendipitous recommendations. In our investigations we consider a proof-of-concept of serendipitous content-based recommendations.

Athena is a content-based recommender that uses a domain ontology, instead of the traditional term-based approach, to recommend news articles [5]. By means of natural language processing, the underlying Hermes framework adds semantic annotations to news articles of Yahoo Finance [10]. These annotations are used to find relations between items in the user profile, which were previously consumed by the user, and unseen items. Five semantic measures are compared to the traditional term-based method. The best performing semantic measure outperforms the term-based measure in all evaluation metrics, i.e. accuracy, precision, recall and specificity. This semantic measure exploits all semantic relationships of concepts that available in the ontology to find indirect links between news articles.

The CHIP Art Recommender exploits the domain-specific ontology of a Dutch museum (Rijksmuseum) to recommend art to visitors of the museum [13]. Next to each recommendation is a button “why recommended?”, which can be clicked to show a reason for the recommendation. This reason is the articulation in natural language of the semantic path that was used to generate the recommendation. The CHIP art recommendations are evaluated by means of two measures: accuracy and interestingness. These measures are chosen for two reasons: (1) it is difficult to determine the total number of relevant items, which is mandatory to calculate the recall, and (2) relevance is subjective and influenced by (the availability of) explanations of recommendations, as discussed in [2].

Feature-based explanations in recommender systems can have a positive influence on the perceived quality of recommendations, because explanations help users understand the commonality between items [12]. Examples of features that are applicable to explaining why two movies are related include genre, actor and director. The user study in [12] shows that feature selection in explanations needs to be tailored to the user, since different users are interested in different features. Also, features selection in explanations should be tailored to the context, e.g. social setting and mood. The authors suggest that users are enabled to manually select a feature that is used in explanations.

The use of the owl:sameAs predicate is not limited to the strict logical semantics that are demanded by its definition. In [3] four distinct (ab)uses of the owl:sameAs predicate are discussed that are different from its intended use, with implicit semantics such as represents and very similar to. In our research here we have not encountered problematic declarations of owl:sameAs links, although we use such identity relations extensively. However, we have observed that in the LinkedMDB dataset there are identity links to Freebase that (ab)use the foaf:page predicate.

3.2. Property-based Recommendations

The recommender service in our project relies on a simple consideration: browsing property relations from both user activity objects and media items we can reach some common nodes. Media items, which lead to a lot of these common points, will be recommended to the user with those activities. So in this case the user profile is not relevant (if we consider the profile something different from user activities). The algorithm devised leverages both inbound and outbound properties from activity objects and media items. Here's an example:
User Jana
- Jana liked George Clooney
- Jana watched 'Death Proof' directed by Quentin Tarantino
- Jana is recommended to watch 'From dusk till dawn' also directed by Quentin Tarantino and starring George Clooney.

For an approach like this, we start by calling a service¹ and getting all resources that are objects of activities performed by a certain user:

```xml
<?xml version="1.0"?>
<activityObjects>
  <resource>http://www.myspace.com/thepillowfactory</resource>
  <resource>http://dbpedia.org/resource/Eiffel_Tower</resource>
  <resource>http://en.wikipedia.org/wiki/Ben_Stiller</resource>
</activityObjects>
```

These URIs are set by different sources: DBpedia, Wikipedia, Last.fm, Myspace and so on. In order to query the cloud of Linked Open Data (LOD) we need one or more identifiers used within the LOD cloud itself, otherwise we can't access the cloud. We chose to start by DBpedia, being the most used and referred dataset in the cloud:

¹ http://www.notube.tv/wiki/index.php/Getting_user_data#Get_user_activity_resources
By simply replacing prefixes or performing specific SPARQL queries, we try to get all DBpedia URIs out from these different kinds of identifiers:

- DBpedia for general domain
- LinkedMDB for movies
- MusicBrainz for bands
- Freebase for general domain (similar to DBpedia but with a controlled authoring and filtering)

In a parallel fashion each of these datasets is queried in order to browse property chains from/to user activity objects and from/to EPG resources. Whenever a common resource is found, and thus a path between an activity object and an EPG resource is established, the EPG media item that owns that resource has an incremented possibility to be recommended to that user.

The algorithm:
After having explained in this subsection the main idea and setup of the property-based recommendation approach used in the NoTube recommendation service, in the next section we go into depth of the experiments performed investigating the effect of different connections between properties, i.e. different patterns.

### 3.3. Linked Data patterns

In the NoTube project we have explicitly experimented with content-based recommendation methods that use Linked Data (LD) as background knowledge for producing novel, serendipitous recommendations. These methods are not meant to replace, but rather to complement existing recommender systems, by focusing especially on the serendipity (diversity) aspects of each result and the result set as a whole. For TV recommendations novelty and serendipity are key, because TV is often a leisure activity and broadcasted media vary in genre and topic. Novelty and serendipity in recommender systems are not well-explored research topics [4] and some believe that content-based methods are inherently incapable of producing unexpected results [7].

Currently much LD is published, but an unsolved problem lies in how to filter this multitude of facts for bits that are relevant for a specific purpose. For example, the fact that *Madonna* is a human being and *not a plant*, is necessary knowledge from a modeling point of view, but not interesting from a consumer’s perspective. One approach to filtering relevant data is by using predefined patterns, which identify relevant path archetypes [13]. Using such patterns in LD enables the system to explain recommendations, by articulating the path between the user interest and the recommended item. Since nodes in LD are assigned labels (as dictated by LD principles) and optionally a depiction (link to an image), semantic paths are suitable to be articulated either graphically or in natural language. Here, we use two patterns, that we selected by manual inspection of LD sources, to generate recommendations. We have set up an experiment where the recommendations are evaluated both in terms of accuracy and novelty. We studied whether and how the novelty factor is what distinguishes LD recommendations from traditional content-based recommendations and state-of-the-art collaborative recommenders.

![Figure 3: Properties-based recommendation algorithm](image-url)
The data we use to produce recommendations is:

- **background knowledge from Linked Data**: DBpedia (which can be called the hub of the LD cloud with the greatest amount of links to other datasets), LinkedMDB (a dataset with a strong focus on movies), and Freebase (which contains accurate and up-to-date information).

- **a user profile**: Social Web activity data aggregated by the Beancounter. More details in D 3.4

- **item (program) metadata enriched with LD**: The item metadata is fetched from EPG metadata sources, such as BBC /programmes. Our NoTube EPG service enriches names of programs and people to provide enriched EPG metadata of both recent and archive listings in JSON format. We enrich strings whenever possible, i.e. we enrich lexical data with relational data; URI’s. For example, when we encounter the string “Magnolia” or “Philip Seymour Hoffman”, we add their corresponding URI’s to the EPG metadata using the NoTube Lupedia service.

In the initial study we used **two path patterns**. Pattern selection was done through manual inspection of LD sources. Both patterns involve background information on people, as the intuition is that the people involved, e.g. directors of movies and guests on talk shows, influence program choice. Movie enthusiasts confirm this. The first pattern (“implicit-participant”) represents a path between two programs through common participants. The second pattern exploits an explicit “influenced-by” link between people.

**“Implicit-participant” Pattern:**

The motivation behind this pattern is the following: when users express an interest in a particular program, such as a movie, they are also potentially interested in the people playing a “role” in this program, without an explicit statement to this effect. A “role” is something like an “actor”, “presenter” or “guest”. We can use background knowledge of people’s “roles” to recommend a different program, in which a person playing a role in the program with an explicit user interest also plays a (different) role. For example,

we recommend an episode of the BBC talk-show “Friday Night with Jonathan Ross”

because the guest is an actor in the movie “Magnolia”

and this person is also in the user’s profile.

Figure 4: Instantiation of “implicit-participant” pattern of type: linkedmdb:actor. The shaded boxes represent media items. The striped boxes indicate where the data resides; the dotted nodes indicate the type of nodes for clarity reasons. depicts this example and displays the exact semantic relations between the entities involved. This pattern exploits the fact that a lot of metadata of participants in movies is listed on the Web. We call this the “implicit-participant” pattern. LinkedMDB provides a number of predicates for identifying roles of people in programs. We selected the five most frequent predicates (see Table 1: Statistics w.r.t. “implicit-participant” pattern: number of triples of all predicates that have a film as subject and a person as object. Variables: ?i is a user interest; ?f is a movie feature that may match a TV show feature. ).
Figure 4: Instantiation of “implicit-participant” pattern of type: linkedmdb:actor. The shaded boxes represent media items. The striped boxes indicate where the data resides; the dotted nodes indicate the type of nodes for clarity reasons.

Table 1: Statistics w.r.t. “implicit-participant” pattern: number of triples of all predicates that have a film as subject and a person as object. Variables: ?i is a user interest; ?f is a movie feature that may match a TV show feature.

<table>
<thead>
<tr>
<th>pattern</th>
<th>nr paths</th>
<th>weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>?i linkedmdb:actor ?f</td>
<td>389,298</td>
<td>1.0</td>
</tr>
<tr>
<td>?i linkedmdb:director ?f</td>
<td>106,360</td>
<td>0.9</td>
</tr>
<tr>
<td>?i linkedmdb:producer ?f</td>
<td>35,388</td>
<td>0.8</td>
</tr>
<tr>
<td>?i linkedmdb:writer ?f</td>
<td>28,649</td>
<td>0.7</td>
</tr>
<tr>
<td>?i linkedmdb:music_contributor ?f</td>
<td>19,643</td>
<td>0.6</td>
</tr>
<tr>
<td>?i linkedmdb:editor ?f</td>
<td>18,490</td>
<td>N.A.</td>
</tr>
<tr>
<td>?i linkedmdb:cinematographer ?f</td>
<td>3,910</td>
<td>N.A.</td>
</tr>
<tr>
<td>?i linkedmdb:film_art_director ?f</td>
<td>2,625</td>
<td>N.A.</td>
</tr>
<tr>
<td>?i linkedmdb:film_story_contributor ?f</td>
<td>1,202</td>
<td>N.A.</td>
</tr>
<tr>
<td>?i linkedmdb:story_contributor ?f</td>
<td>1,202</td>
<td>N.A.</td>
</tr>
<tr>
<td>?i linkedmdb:executive_producer ?f</td>
<td>954</td>
<td>N.A.</td>
</tr>
<tr>
<td>?i linkedmdb:film_production_designer ?f</td>
<td>581</td>
<td>N.A.</td>
</tr>
<tr>
<td>?i linkedmdb:film_casting_director ?f</td>
<td>542</td>
<td>N.A.</td>
</tr>
<tr>
<td>?i linkedmdb:film_set_designer ?f</td>
<td>461</td>
<td>N.A.</td>
</tr>
<tr>
<td>?i linkedmdb:costume_designer ?f</td>
<td>32</td>
<td>N.A.</td>
</tr>
</tbody>
</table>

“Influenced-by” Pattern:

The second pattern we consider is based on the motivation that if the user has expressed an interest in a person, she might also enjoy watching a TV program that features someone else who has influenced, or is influenced by, that person. We can use such background knowledge on “influence” relationships to recommend programs which feature people that have (had) some influence relation with a user’s interest, either directly or via a third entity (i.e. a person in-between). For example,

if the user has interest in Bill Hicks (a comedian from the eighties)
we can recommend a TV program featuring Russell Brand (contemporary comedian)
because “Russell Brand is inspired by Bill Hicks” (from LD sources).

For this pattern we use the predicates dbpedia-owl:influencedBy and dbpedia-owl:influenced from the DBPedia dataset. These predicates are generally used to define influences between people, with some exceptions, such as “Suzana Ansar is influenced by Bangladesh”. Table 2 shows the amount of paths using both influence-relations. Figure 2 depicts the “influenced-by” pattern graphically. DBpedia provides a significant set of such “influenced-by” relations. This set can be expanded with similar relations in other datasets, such as Freebase. Combining these metadata sets requires duplicate detection.

Table 2: Influenced-by pattern. Variables are the same as in Table 1 and ?x is an intermediate entity

<table>
<thead>
<tr>
<th>pattern</th>
<th>nr paths</th>
<th>weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>?i (dbpedia-owl:influencedBy—dbpedia-owl:influenced) ?f</td>
<td>18,338</td>
<td>1.0</td>
</tr>
<tr>
<td>?i (dbpedia-owl:story—who{dbpedia-owl:influenced} ?f</td>
<td>336,443</td>
<td>0.7</td>
</tr>
</tbody>
</table>

Figure 5: Instantiation of “influences-by” pattern. The shaded boxes represent media items. The striped boxes indicate where the data resides; the dotted nodes indicate the type of nodes for clarity reasons.

**Recommendation score:**

To calculate the score of a particular recommendation the score of the pattern matches are summed up. The score of a pattern match is the product of the pattern weight (as listed in Figures 4 and 5) and the weight of the user interest. The weights of the patterns are chosen ad hoc using common sense. To fine-tune the recommendation scores, these pattern weights could be determined using an appropriate methodology, e.g. by asking many movie enthusiasts to assign a weight to each role and using the average answer.

**Evaluation metrics:**

As addressed above, the recommendations we aim to produce deviate from traditional recommendations. By exploiting background knowledge, and articulating this knowledge in explanations, we aim to produce novel and serendipitous recommendations. These qualities are not taken into account by traditional recommender systems evaluation measures, such as precision and recall [13]. To evaluate the usefulness of the selected patterns, we set up a user evaluation experiment. In this experiment users were asked to provide feedback on the accuracy and novelty of recommendations. Here we tested the quality of the recommendation strategy independently from the performance of the user-profile service. In order to do so, we presented the evaluators a set of recommendations generated using a selected topic of interest. Evaluators could choose from 15 topics of interest, of which 8 are people and 7 are programs. Evaluators were requested to evaluate recommendations of 6 topics, preferably 3 people and 3 programs, and were presented with 5 recommendations for each topic of interest. Figure 6 is a screenshot of a single recommendation in the evaluator Web interface.
Figure 6: An example of a recommendation in the evaluation interface

Of each recommendation, we display the program title, explanation of the recommendation, program metadata and a depiction that is part of the original EPG data. The program metadata consists of the title, genre, synopsis and the person’s role in that TV program. Links to metadata on Wikipedia are added so users can easily navigate to external information about the matched person and TV program. The two evaluations question we asked were “Given the topic of interest, is this a reasonable suggestion?” and “Would you consider this recommendation trivial or novel?”. Answers to the questions were on a scale from 1 to 5, where 1 is “terrible” and “boring”, respectively, and 5 is “fantastic” and “pleasantly surprising”. Evaluators could also choose to answer that they do not know how to evaluate this particular recommendation. For each question, evaluators could optionally add comments in a text box. In total 20 evaluators rated 485 recommendations. The evaluators were a diverse group with ages ranging from 24 to 67, six different nationalities and seven different occupations.

Results:
The overall averages of the evaluations for both patterns are shown in Table 3. We see mediocre average outcomes for both patterns, which leaves us with two questions: are the questions answered mediocre in general or do the answers vary? And: how do the answers of the different variants of the patterns relate?

Table 3: The average accuracy and novelty of both patterns and the amount of times evaluators indicated they did not know how to evaluate a recommendation
The histograms in Figure 7 and 8 show the percentage of answers of every pattern type below.

**Results for pattern “implicit participant”**: this pattern is subdivided in five types, where each type corresponds to an applied predicate (see Table 1). The histograms mentioned above show that the results of the “implicit participant” pattern vary. For example, if we consider the evaluations in Figure 7(a) of type `linkedmdb:music_contributor`, we see that 33% was considered terrible (answer 1) and another 33% was considered good (answer 4). We can also see that recommendations using this predicate are considered novel, because Figure 7(a) shows us that the majority of evaluators was positive in their answers to the novelty-question (answer 4).

**Results for the pattern “influenced by”**: for this pattern we distinguish two types: a direct and indirect match. We speak of a direct match when there is a single influence-relation between a user interest and participant of a TV program. An indirect match occurs when two influence-relations link a user interest and a TV program participant via a person-in-between. We see that the results are more consistent than that of the “implicit participant” pattern. Figure 8(b) shows that especially the novelty factor scores are high: the majority of the evaluators answered 4. As expected, we see that the quality of the direct “influenced by” matches is greater than that of the indirect matches, since the semantic distance between the user interest and the recommended TV program is greater for indirect matches. We did not expect such a small difference between the direct and indirect matches, which shows that the extra information that the indirect match provides, is considered interesting by the evaluators.
Co-occurrences of answers: To gain insight in the dynamics between the accuracy and novelty evaluations, we have generated co-occurrence matrices of the answers in the two different patterns (Table 4 and 5). The co-occurrence matrix of the “implicit participant” pattern (Table 4) has the highest values on the diagonal. This tells us that there is a strong correlation between the answers to the accuracy and novelty questions. In the co-occurrence matrix of the “influenced by” pattern (Table 5), we see the highest values in the center. This is mainly due to the majority of the answers being either 3 or 4 as stated above. Also, 10% of the “influenced by” recommendations are evaluated with 1 and 4 for the accuracy and novelty respectively. This shows that many recommendations were considered inaccurate, but the extra information provided in the explanation was interesting nonetheless.

Table 4: Accuracy/novelty distribution table of the « implicit-participants » patterns. The different accuracy evaluations are distributed over the columns. The different novelty evaluations are distributed over the rows

Table 5: Accuracy/novelty distribution table of the «influenced-by» patterns. The different accuracy evaluations are distributed over the columns. The different novelty evaluations are distributed over the rows

Insights from feedback: From reading the comments and other feedback from evaluators, we observed that the pattern type linkedmdb:actor often gives trivial results, due to a lack of a so-called billing order (an ordering of the actors by importance in that movie or TV show). For example, the movie Mimic was recommended, because it features Josh Brolin, although he is the 6th on the list of most important actors.
A second observation is that there are a select number of highly influential people, who often appear as the person-in-between in a “influenced by” path. For example, Chevy Chase influenced many people, one of which is Will Ferrell. As a result, when the user has an interest in Will Ferrell, many paths via Chevy Chase show up in the recommendations. To prevent this we will take the diversity with respect to the explanations into account in future prototypes.

In the comments we saw that evaluators were disappointed when there was a simpler reason for recommending an item than the presented explanation. For example, with Will Ferrell as the user interest the movie Blades of Glory was recommended. Although Will Ferrell is himself featured in this movie, the explanation of the recommendation is “This item features Will Arnett, who is influenced by Chevy Chase, who in turn influenced Will Ferrell”. The more direct explanation that Will Ferrell is lead actor in this particular movie is more appropriate here.

**Discussion on recommendations with LD patterns:**

From these results we can try to understand the effort and effect to use Linked Data for recommendations in the TV domain. Current recommenders are mainly based on collaborative or statistical techniques. Content-based recommenders are not meant to replace such recommenders. Rather, our hypothesis is that content-based recommendations can be complementary: proposing related programs through semantic links, which are not part of the “regular” suggestions. Content-based suggestions will not always be correct: if a considerable number is judged as “surprising” or “novel”, we have achieved our purpose. A strong point of content-based recommendation in general is that it provides an explanation of why the recommendation was made, as shown in this paper. This feature may be in itself a powerful supplement to other recommender methods.

For generating the recommendation we used path patterns in Linked Data. Given the size of the search space, we view identification of useful patterns as a key topic in semantic search in general, of which content-based recommendation can be viewed as a special case. We are aware that with experiments like these we have been able to only start scratching the surface of pattern-based search. Much more (empirical) work is needed in this area of semantic search, and for its application in this domain of TV recommendations in particular.

The results of the user study show that “interestingness” is highly subjective. Users judge the same suggestions differently. This is to be expected and should not be considered a bug. If a significant number of users found a recommendation interesting we should view this as a positive sign.

In the “implicit-participant” pattern the “actor” relation was less effective compared to others such as the “writer”. The assumption behind the pattern is that users will not remember the names of people involved in a movie they like, and will therefore be pleasantly surprised with suggestions of other programs involving these people. Apparently, the subjects knew the names of the actors quite well, so were not surprised seeing them in the recommendations (and therefore scored lower on “novel”). The “influenced-by” pattern appeared to generate better results than the “implicit-participant” pattern. A plausible explanation for this is that this pattern puts relationships between people at the heart of the recommendation. Such relationships might be inherently more interesting: one can view it maybe as semantic gossiping.

This limited scope of these experiments warrants no general conclusions about the usefulness of content-based recommendations using Linked Data, but it definitely shows that this is a hypothesis worth further exploration.
4. Statistics-based Recommendation Strategies

In NoTube we have been making use of the Apache Mahout toolkit, which provided us with software for collaborative filtering recommendations, clustering and automatic classification. We have barely scratched the surface of what it can do, but here show some initial results applying Mahout to a 100,000 record subset of Harvard’s 14 million entry catalogue. Mahout is built to scale, and the experiments here use datasets that are tiny from Mahout’s perspective.

In NoTube, we used Mahout to compute similarity measures between each pair of items in a catalogue of BBC TV programmes for which we had privileged access to subjective viewer ratings. This was a sparse matrix of around 20,000 viewers, 12,500 broadcast items, with around 1.2 million ratings linking viewer to item. From these, after a few rather-too-casual tests using Mahout’s evaluation measure system, we picked its most promising similarity measure for our data (LogLikelihoodSimilarity or Tanimoto), and then for the most similar items, simply dumped out a huge data file that contained pairs of item numbers, plus a weight.

Alternatively, we could have made use of additional metadata published by the BBC in RDF, so we can help out Mahout by letting it know that when Alice loves item_62 and Bob loves item_82127, and via RDF we also know that they are both in the same TV series and Brand (according to the BBC Programme Ontology). But why use fancy machine learning to rediscover things we already know (or that have been shared in the Web as data)? We could make smarter use of metadata here. Secondly, we could have used data-derived or publisher-supplied metadata to explore whether different Mahout techniques work better for different segments of the content (factual vs fiction) or even, as we have also some demographic data, different groups of users. Anyway, Mahout gave us item-to-item similarity measures for TV. We have shown how we used these in ‘second screen’ (or ‘N-th’ screen, aka N-Screen) prototypes demonstrating the impact that new Web standards might make on tired and outdated notions of “TV remote control”.

What if your remote control could personalise a view of some content collection? What if it could show you similar things based on your viewing behavior, and that of others? What if you could explore the ever-growing space of TV content using simple drag-and-drop metaphors, sending items to your TV or to your friends with simple tablet-based interfaces?

There are NoTube prototypes using BBC content (sadly not viewable by everyone due to rights restrictions), but also some experiments with TV materials from the Internet Archive, and some explorations that look at TED’s video collection as an example of Web-based content that (via ted.com and YouTube) are more generally viewable. Since every item in the BBC’s Archive is catalogueued using a library-based classification system (Lonclass, itself based on UDC) the topic of cross-referencing books and TV has cropped up a few times.

Meanwhile, in (the digital Public Library of) America, the Harvard Library Innovation Lab team have a huge and fantastic dataset describing 14 million bibliographic records. We have been trying to help figure out how we could cross-reference their records with other “Webby” sources, such as online video materials. Again using TED as an example, because it is high quality but with very different metadata from the library records. So we’ve been looking at various tricks and techniques that could help us associate book records with those. So for example, we can find tags for their videos on the TED site, but also on delicious, and on YouTube. However taggers and librarians tend to describe things quite differently. Tags like “todo”, “inspirational”, “design”, “development” or “science” don’t help us pin-point the exact library shelf where a viewer might go to read more on the topic. Or conversely, they don’t help the library sites understand where within their online catalogues they could embed useful and engaging “related link” pointers off to TED.com or YouTube.

3 http://mahout.apache.org/
4 http://www.bbc.co.uk/programmes/
5 http://notube.tv/2011/10/10/n-screen-a-second-screen-application-for-small-group-exploration-of-on-demand-content/
So we turned to other sources. Matching TED speaker names against Wikipedia allows us to find more information about many TED speakers. For example the Tim Berners-Lee entry, which in its Linked Data form\(^6\) helpfully tells us that this TED speaker is in the categories ‘Japan Prize laureates’, ‘English inventors’, ‘1955 births’, ‘Internet pioneers’. All good to know, but it’s hard to tell which categories tell us most about our speaker or video. At least now we’re in the Linked Data space, we can navigate around to Freebase, VIAF and a growing Web of data-sources. It should be possible at least to associate TimBL’s TED talks with library records for his book\(^7\) (so we annotate one bibliographic entry, from 14 million!). Can we do better? What if we also associated Tim’s two TED talk videos with other things in the library that had the same subject classifications or keywords as his book? What if we could build links between the two collections based not only on published authorship, but on topical information (tags, full text analysis of TED talk transcripts). Can we plan for a world where libraries have access not only to MARC records, but also full text of each of millions of books? Dan Brickley has been exploring some of these ideas with David Weinberger, Paul Deschner and Matt Phillips at Harvard, and in NoTube with Libby Miller, Vicky Buser and others.

The big story is in linking TV materials to the gigantic back-story of context, discussion and debate curated by the world’s libraries. If we can imagine a view of our TV content catalogues, and our libraries, as visual maps, with items clustered by similarity, then NoTube has shown that we can build these into the smartphones and tablets that are increasingly being used as TV remote controls. And if the device you’re using to pause/play/stop or rewind your TV also has access to these vast archives as they open up as Linked Data (as well as GPS location data and your Facebook password), all kinds of possibilities arise for linked, annotated and fact-checked TV, as well as for showing a path for libraries to continue to serve as maps of the entertainment, intellectual and scientific terrain around us.

\(^6\) http://dbpedia.org/page/Tim_Berners-Lee
\(^7\) http://openlibrary.org/books/OL38986M/Weaving_the_Web
The images above were made with Gephi, Mahout and experimental data from the Library Innovation Lab at Harvard, plus a few scripts to glue it all together. Mahout was given 100,000 extracts from the Harvard collection (sub-title, a local ID, and a list of topical phrases mostly drawn from Library of Congress Subject Headings, with some local extensions). They are treated as atomic codes, and flattened into long pseudo-words such as ‘occupational_diseases_prevention_control’ or ‘french_literature_16th_century_history_and_criticism’, ‘motion_pictures_political_aspects’, ‘songs_high_voice_with_lute’, ‘dance_music_czechoslovakia’. All of human life is there. David Weinberger has been calling this gigantic scope our problem of the ‘Taxonomy of Everything’, and the label fits. The result is a matrix of 100,000 bibliographic entities, by 27684 unique topical codes. Initially we made the simple test of feeding this as input to Mahout’s K-Means clustering. Manually inspecting the most popular topical codes for each cluster (both where k=12 to put all books in 12 clusters, or k=1000 for more fine-grained groupings). See some results8 9 and the dictionary10.

For example, in the 1000-cluster version, we get:

'medical_policy_united_states', 'health_care_reform_united_states', 'health_policy_united_states',
'medical_care_united_states', 'delivery_of_health_care_united_states', 'medical_economics_united_states',
'politics_united_states', 'health_services_accessibility_united_states', 'insurance_health_united_states',
'economics_medical_united_states'.


'collectors_and_collecting_history'. And another, nearby ‘art_thefts’, ‘theft_from_museums’,
'archaeological_thefts', ‘art_museums’, ‘cultural_property_protection_law_and_legislation’, ...

This shows that the tooling is capable (by looking at book/topic associations) at picking out similarities that are significant. (A side-goal here is to publish these clusters for re-use elsewhere). It shows that if we can find a way to open up bibliographic datasets, there are solid opensource tools out there that can give new ways of exploring the items described in the data. That those tools (e.g. Mahout, Gephi) provide many different ways of computing similarity, clustering, and presenting. There is no single ‘right answer’ for how to present literature or TV archive content as a visual map, clustering “like with like”, or arranging neighbourhoods. And there is also no restriction that we must work dataset-by-dataset, either. Why not use what we know from movie/TV recommendations to arrange the similarity space for books? Or vice-versa?

8 http://danbri.org/2011/mahout/hv/_k1000.txt
9 http://danbri.org/2011/mahout/hv/_twelve.txt
10 http://danbri.org/2011/mahout/hv/_harv_dict.txt
This work was done as a proof-of-concept, which shows some potential, but it is neither a user interface, nor particularly informative. Gephi as a tool for making such visualizations is powerful, but it too is not a viable interface for navigating TV content. However these tools do give us a glimpse of what is hidden in giant and dull-sounding databases, and some hints for how patterns extracted from these collections could help guide us through literature, TV or more.

There are many things that could be tried; we would like to explore some variant of these 2D maps onto ipad/android tablets, loaded with TV content. We would like to continue exploring the bridges between content (e.g. TED) and library materials, on tablets and PCs. We would like to look at Mahout’s “collocated terms” extraction tools in more details. These allow us to pull out recurring phrases (e.g. “Zero Sum”, “climate change”, “golden rule”, “high school”, “black holes” were found in TED transcripts\(^{11}\)). We also tried extracting bi-gram phrases from book titles\(^ {12}\) using the same utility. Such tools offer some prospect of bulk-creating links not just between single items in collections, but between *neighbourhood regions* in maps such as those shown here.

As full text access to book data looms, and TV archives are finding their way online, we’ll need to find ways of combining user interface, bibliographic and data science skills if we’re really going to make the most of the treasures that are being shared in the Web.

We conclude this section with an observation. A few years ago, Netflix\(^ {13}\) had the vision and cash to pretty much buy the attention of the entire machine learning community for a measly million dollars. Researchers love to have substantive datasets to work with, and the (now retracted) Netflix dataset is still widely sought after. Without a budget to match Netflix’, could we still somehow offer prizes to help get such attention directed towards analysis and exploitation of linked TV and library data? We could offer free access to the world’s literature via a global network of libraries? Except everyone gets that for free already. Maybe we don’t need prizes.

### 5. Explanations and User Feedback

An important element of recommendation quality has been explanations. Explanations of recommendations have shown to improve trust of users, and thus the adaptation of recommender systems [11]. Also, explanations can contain interesting information that the user may be unaware of. For these two reasons, our recommender prototype generates explanations of recommendations. The Semantic Web data that are used as background information by the recommender is particularly suitable for generating such explanations, because Linked Data (LD) principles dictate that nodes should have associated labels. We use these labels to generate natural language explanations. For example,

> *This talk show features Philip Seymour Hoffman as an interviewed guest, who also acts in the movie Magnolia, which you watched last week*.

The emphasized words in this explanation are labels from entities as defined in LD. We could also show the portraits of Bill Hicks and Russell Brand connected by an arrow that contains the words “influenced-by” to illustrate the relation between these comedians.

Research shows that accompanying recommendations with explanations helps users to make more accurate decisions, improves user acceptance of recommendations, and increases trust. A survey of users of one movie recommender study showed that 86% of those surveyed wanted an explanation feature added to the site (Herlocker et al, 2000). Similarly, users who understand why an item is being recommended reported a higher degree of confidence, liking and understanding (Sinha 2002).


\(^{13}\) [http://www.netflixprize.com/](http://www.netflixprize.com/)
According to Bilgic and Mooney (2005) "a good explanation is one which accurately illuminates the reasons behind a recommendation and allows users to correctly differentiate between sound proposals and inadequately justified selections."

5.1. Rationale for Explanations of Recommendations

Even if a Linked Data based recommendation scores poor results in a traditional recommender system evaluation, a quirky story backing it can make it interesting, leading to greater user satisfaction. Presenting the pathways through the graphs allows the user to see the connection that led to a recommendation being made. This gives the user a more interesting story than the 'black box' explanations of collaborative filtering techniques, such as "You might like Y because other people who liked X also liked Y". We also assume that different types of connections will trigger different levels of interestingness in the explanations, resulting in different levels of user satisfaction. For example, links based on subject may be generally more or less interesting than links based on people (such as actors, directors, authors or presenters).

The main challenges we have identified with respect to explanations of recommendations consider "How to translate the recommendation algorithms into nuanced, human-like, natural language explanations?". Subsequently, guiding questions we have used for the user studies are:

- Does the presence of an explanation have an effect on user satisfaction regardless of the type of recommendation?
- Does the type of recommendation have an effect on user satisfaction regardless of the presence of an explanation?
- Might there be interactions between the effects of type of recommendation and the presence of an explanation? For example, does the explanation alter the user's satisfaction depending on the type of recommendation?
- Do people find the personalization of the explanation disconcerting?

In their paper Tintarev and Masthoff (2007) examined potential 'features' that might be of interest to users in explanations of movie recommendations - e.g. actors, awards, other people's ratings. Based on this we identified for NoTube features of explanations to be included:

- **Genre**, e.g. comedy, drama
- **Subject matter**, classification terms, keywords
- **People/Contributors**, e.g. director, actor, writer

Tintarev and Masthoff's work on explanations for movies suggests that these features are important for people. Also a glance through reviews for TV-related DVDs on Amazon UK suggests that people frequently refer to writers, actors, comedians and directors when explaining why they like a particular series. Additionally, reviews also often refer to other series in the same genre for comparison - e.g. Downton Abbey is compared favorably to Upstairs Downstairs. From related work in museum artwork recommenders, it appears that **style, creator and human relationships between people** (e.g. student of/teacher of/collaborator) were strong reasons for interest in users and support serendipity.

Additionally, some other features can be included, depending on personal user’s activities and social network, as well as statistical properties of programs:

- Personal activities ("you watch lots of crime dramas")
- Friends' activities
- Program popularity

Further, it is also important to consider the **order in which features are mentioned** and the **optimal number of features to mention** in one explanation.

- the 'features' that are included (genre, director, topic etc) - which are most interesting?
- the order in which the features are presented - does this make a difference?
- the optimal number of features - is there a cut-off point?
- the effects of the size of the graph (i.e. number of pathways making the connections)
How to measure effectiveness of these explanations? The effectiveness of an explanation system can be measured using two fundamentally different approaches: the promotion approach and the satisfaction approach. For the promotion approach, the best explanation is the one that is most successful at convincing the user to adopt an item. For the satisfaction approach, the best explanation is the one that lets the users assess the quality of the item the best. We believe that satisfaction is more important than promotion. If the users are satisfied with their selections in the end, they will develop trust in the system and continue to use it. We have used three explanation systems in our study: keyword style explanation (KSE), neighbor style explanation (NSE), and influence style explanation (ISE). Bilgic and Mooney (2005) found that keyword style explanations, which present content information about an item that caused it to be recommended, or influence style explanations, which present ratings previously provided by the user that caused an item to be recommended, were found to be significantly more effective at enabling accurate assessments than neighbor style explanations.

5.2. Experiment with the Notion of Serendipity

One of our more recent NoTube demos was inspired by an XKCD cartoon\textsuperscript{14} which prompted us to ask: can we make a large video collection interesting enough so that people keep browsing rather than give up? As we have said before\textsuperscript{15}, many people find choosing in large collections difficult. Although there is clearly a place for search in finding known-item video content, the user has to know what they are looking for.

The aim of the demo is to surface niche and diverse programs of interest buried in the ‘long tail’. It is designed to aid serendipitous content discovery by helping people to browse better by following interesting connections between programs using content-based links. In this respect, it is the polar opposite to choosing what to watch based on what is popular. The connections are created using a BBC-specific subject classification system, traditionally used by professional cataloguers for subject indexing/classification as part of the cataloguing procedure in the BBC’s internal TV and radio program catalogue. The classification system reflects the last 50 years of BBC programming: it is extremely granular, TV-centric and includes numerous quirky terms.

For the purposes of the demo, we are using the classification system as a form of Linked Data\textsuperscript{16}. From any program the user can follow suggestions for similar programs based on the number of common classification links between them. The idea is to support ‘hours of fascinated clicking’ through the video collection, similar to the way that following links in Wikipedia articles can take users on surprising and unexpected journeys through the content.

The technique used in the demo generates a similarity measure based on the number of categories in common between any pair of programs and then displayed these to the user as suggestions for related programs.

5.3. Experiment with Explanations and Related Programs

As discussed in section 4, we showed the potential role of serendipity in bringing people’s attention to ‘surprisingly good’ content that they didn’t already know about. Further, we wanted to find out if the experience of serendipitous content discovery could be supported by re-using the metadata created when BBC programs are archived (in this case terms from the BBC subject classification system). We chose a technique based on a similarity measure\textsuperscript{17} based on the number of categories in common between any pair of programs, and then displayed these to the user as suggestions for related programs. We have performed this experiment in the NoTube Archive Browser. For the purposes of this evaluation, we identified two specific research questions. Does it help people find more interesting programs if:

1. they browse similar programs rather than a random selection

\textsuperscript{14} http://xkcd.com/214/
\textsuperscript{15} http://notube.tv/2010/11/19/experimenting-with-linked-data-to-improve-the-tv-experience/
\textsuperscript{16} http://notube.tv/research-topics/linked-data/
\textsuperscript{17} http://en.wikipedia.org/wiki/Jaccard_index#Tanimoto_Similarity_and_Distance
2. they see browsable **subject category information** about the programs

Our hypothesis was that seeing both similar programs and clickable subject category information (as shown in the screenshots below) would be optimal for finding interesting new programs.

![Figure 9: Related programs in NoTube Archive Browser](image_url)
During the trial the 96 participants had seven days during which to browse the Web-based on-demand video collection and to add programs of interest to a playlist. The participants were randomly allocated one of four experimental conditions:

- Random programs, no subject categories displayed (our control)
- Random programs, with subject categories displayed for each
- Similar programs, with no categories displayed
- Similar programs, with subject categories displayed for each (our optimal condition?)

We measured the length of the playlists and the time spent browsing as indicators of ‘interestingness’, based on the assumption that people would create longer playlists and spend more time browsing when they found the links interesting. At the end of the trial 20 of the participants also filled out an online questionnaire about their experiences.

Statistical analysis of the data we collected showed no clear effects of program similarity (fig. 11a) or the display of subject categories (fig. 11b) on the length of participants’ playlists or the time they time spent browsing. Likewise, as the graphs below illustrate, responses to the questionnaire showed no significant variation in people's experiences between the four experimental conditions. However, the questionnaire results suggest that, regardless of the experimental condition, people did find interesting new programs and generally liked the application.

For example, one participant remarked: “I liked the way the app threw up programs I had forgotten about. I liked the way that on selecting one program I always found even more interesting programs
on the related programs list." In total, 15 out of the 20 respondents said they would recommend the application to others.

![Figure 12: Comparison of questionnaire results between control and optimal groups](image)

One possible explanation for the fact that the experimental condition had no significant effect on people’s enjoyment of the application was that our ‘random’ selection was actually too good, since it tended to display a fairly reasonable set of programs taken from a variety of well-known TV series.

Overall, the results suggest to us that adopting this type of approach to navigating large video collections (including archives, on-demand content and EPGs) could help with the ‘cold start’ problem, whereby a recommendation system cannot provide useful recommendations until a substantial amount of user activity data has been gathered. Re-using existing program metadata to create content-based links in this way could help large media organizations to provide users with a means of accessing niche video content that might not otherwise be found, in particular if no-one has yet watched it because it hasn’t appeared in a ‘What’s popular’ or ‘What’s new’ list.

6. Deployment in NoTube Use Cases

6.1. Semantic News Recommendations (WP7a)

This personalised newscast scenario focuses on the design and development of a system for the creation and the delivery of a set of local personalised news services. The recommendation service is based on the matching between the user profile analysis coming from the Beancounter and the concepts extraction coming from the analysis of the speech to text done with Lupedia. This matching gives a ranking used to personalise news items for a particular user. WP3’s user activity logging (Beancounter) and news items recommendation services are integrated into the WP7a prototype via WP6 APIs provided by the RESTful back-end services of the NoTube User Portal (see Figure 13).
6.2. Adaptive Ad Inclusion (WP7b)

The personalized ad algorithm works as a property weighted algorithm, where different properties of the user profile and ad metadata have an influence on the final result. This influence can be adjusted by means of the weight which tells the algorithm how heavy a certain property should be taken into account. The main properties that we consider in this case can be classified in two groups:

1. Demographical properties: Age, gender, education, lifestyle, location.
2. Program/metadata properties: Genre, keywords, semantic distance.

Considering all these properties and their respective weights, the algorithm calculates a score for every ad in the batch of currently valid ads. This score can then in turn be used to rank the ads, telling us which ad currently fits the user's profile best.

6.3. Recommendations in KT demonstrator (WP7b)

“KT olleh tv now” is an IPTV service for mobile smart devices and lets users watch TV & VOD on the road. Developed by KT, it has 30 live channels and 5000 VODs. In order to help users choosing what VODs to watch, it recommends a list of VODs that the users might like to watch based on their previous watching behaviours. Currently, the recommendation is mainly based on a collaborative filtering and it first builds a (VOD content - VOD content) matrix determining the relations between
pairs of contents. Using this matrix and the watching behaviour of a current user, the recommendation engine can predict what VOD would the user watch next. It is similar to the Amazons’ “users who bought x also bought y” type of recommendation.

6.4. Recommendations in BBC demonstrator (WP7c)

One of the main themes in NoTube is connecting the Web and TV through the use of Linked Data\textsuperscript{18} to enhance the TV experience. In this part of the project we are specifically interested in exploring how Linked Data can be used to help people:

1. **Decide what to watch:** With so much choice available, it’s becoming harder to decide what to watch. We are therefore looking at ways of filtering programmes of interest and bringing them to the user’s attention.

2. **Find out more about a programme:** Currently this involves having to remember to look for the relevant information once the programme has finished, or interrupting the programme to look for it now, with the risk of missing something else of interest in the process.

3. **Have smarter online conversations about TV:** There is no shortage of online conversation about TV: witness how frequently Twitter trending topics are TV-related during peak-time viewing (at least this is often the case in the UK). However, Web-based conversations, about TV or anything else, cannot be held without URLs that allow people to refer to the thing they are talking about.

Key to all of these goals is the disambiguation of specific program episodes with unique IDs. For example, the BBC’s /programmes website provides persistent links to program episodes from which machine-processable programs metadata can be accessed. Resolving a /programmes URL (such as the URL for an episode of Eastenders\textsuperscript{19}): gets you data about the program, available in various formats including RDF, such as when it’s on next, whether it’s available to watch on iPlayer, a description of it, whether it was part of a series or not, what version it was (signed, shortened), and sometimes who was in it, what role they played, who the director and writer were, and so on. These URLs provide the starting points for linking out from the program episode to related information from the Linked Open Data cloud, adding background knowledge and context to the program and making new, and sometimes surprising, connections.

In **helping people to decide what to watch**, we hypothesize that some of these types of connections will be of particular interest to people. The approach assumes that filtering and presenting the links that led to these connections will provide useful information about whether a program is worth watching or not. For example: “This program stars [your favorite actor] and/or is about [your town] and/or is about [a topic of particular interest]”. This approach can also surface programs of interest buried in the long tail (i.e. in a large collection of on-demand video content), because it does not rely on lots of people having watched something before it can be suggested.

In **helping people to find out more**, using the program URL to automatically link out to related information on the Web from the Linked Open Data cloud allows relevant background information about a program to be ‘pushed’ to the user in a convenient and unobtrusive way, without the user having to search for anything. For example, an early NoTube demo showed the Wikipedia page about a costume drama, The Forsyte Saga\textsuperscript{20}, being sent to the user’s smartphone whilst they were watching the program on TV. The Wikipedia page provides detailed information about the Forsyte Saga books, the author, TV and film adaptations, main characters, plot, and so on. The user could quickly bookmark the information for later if they didn’t want to interrupt the program by reading it now.

In **supporting online conversations about TV**, the URLs create linkage points into social network discussions, as well as shared bookmarking and commenting systems, enabling the video to be

\textsuperscript{18} http://notube.tv/research-topics/linked-data/
\textsuperscript{19} http://www.bbc.co.uk/programmes/b00vzxms
\textsuperscript{20} http://en.wikipedia.org/wiki/The_Forsyte_Saga
referenced permanently as a focal point that can be annotated. The aim is to extend this by providing unique identifiers (based on URLs) for particular events within an episode. This would allow people to reference specific sequences or moments in a program, for example when discussing “that goal”, “that outrageous comment” or “that hilarious sketch”.

We are currently in the process of planning some user evaluation studies which we hope will validate our hypotheses concerning the advantages of using Linked Data to support these user goals. In the meantime, much of the technical effort is going into setting up the required infrastructure by using URLs to uniquely identify programmes. This is crucial for creating greater coherence for machines, and thereby for TV viewers, who we hope will benefit by enjoying an improved experience.

7. Implementations and Recommendation APIs

7.1. Implementation of the Property-based recommender

This recommender service\(^\text{21}\) matches the Dbpedia Categories (DbpC) of items with a User Profile (UP). An item becomes a recommendation when the DbpC of the item matches a user's interest, i.e. it is an element of the UP. A pair of DbpCs match when they are the same or when they are connected in the DbpC hierarchy through a path of length 3 or less. The DbpCs hierarchy is a tree of skos:broader (and implicitly skos:narrower) links.

**API Parameters** (UP): (type: URI, User Profile, uri of the user profile, i.e. a link to the Beancounter with a user id), example: http://moth.notube.tv:8080/social/rest/people/1238

**API Parameters** (EPG): (type: URI, Electronic Program Guide): defines the service that's called to fetch the EPG from which broadcasts are recommended: http://eculture2.cs.vu.nl:53021/enrichEPG

**Output format**: JSON, an object 'recommendation' with as value a list of objects, which have the following string/value pairs:

- URI (URI of broadcast)
- broadcasts (metadata of the broadcast, generally a uri, pid (programme id), title, start, end, channel)
- recommendationScore (a number between 0 and 1)
- reasons (a list of reasons why this broadcast has been recommended; contains your interest (entry in the user profile), a feature of the broadcast and a relation between the two).

7.2. Implementation of LOD patterns recommender

To demonstrate the pattern-based recommendations and to also test them, we have implemented a recommender prototype that produces JSON output based on three input parameters:

- LD background information
- a user profile
- a set of items (for recommendation purposes).

This recommender prototype has been developed using the ClioPatria framework\(^\text{22}\), an RDF store and HTTP server built on top of SWI-Prolog\(^\text{23}\).

7.3. Implementation of N-Screen recommendation demonstrator

We currently have three different versions of N-Screen running:

- (partial) Redux (http://nscreen.notu.be/redux/)
- TED talks (http://nscreen.notu.be/ted/)
- iPlayer (http://nscreen.notu.be/iplayer/)

---

\(^{21}\) http://eculture2.cs.vu.nl:53020/skosRecommender

\(^{22}\) http://cliopatria.swi-prolog.org

\(^{23}\) http://swi-prolog.org
They all have the same basic design with small tweaks for image size. They interoperate – you can drag and drop between them. The main differences lie in the collection of data for the backends and in the calculation of similarity between videos. For similarity calculation we use three different techniques for the three different datasets, as we had three different sets of data available.

The Redux version was our first experiment. BBC Redux is a BBC research video on-demand testbed. We were lucky enough to be able to obtain anonymised watching data for programs in a five-month subset of the period it covers. So our first experiment was to take that watching data – around 1.2 million observations over 12,000 programs – and use open source tools to generate similarity indexes. We were able to use a standard function in Mahout Apache’s machine learning and data mining software, to generate similarity indexes using a Tanimoto Coefficient model. This function essentially uses people as links between programs (“Bob watched both ‘Cash in the Attic’ and ‘Bargain Hunt’”), and sorts program pairs according to how many people watched them both. With this dataset, this technique produced some nice examples of clearly related clusters (for example what you might call ‘daytime home-related factual’, see picture below).

It is quite rare to have access to this kind of data about what people have watched. It’s both valuable and private, and may not be readily available. It may not exist, if no-one has watched anything yet. For the TED dataset we therefore took a different approach. TED talks are a diverse set of talks by people prominent in their field, licensed under the Creative Commons BY-NC-ND license. From our point of view, the advantage of using this dataset was that transcripts were available for all talks. To calculate similarity between the talks for N-screen we were therefore able to use a tf-idf algorithm. This technique treats each program as a document, and finds the most characteristic words for each document within the total corpus of documents, and can be used to match the documents based on the words selected. We were lucky enough to be able to use some Ruby software open sourced by a colleague at the BBC to do this.

![Figure 14: A cluster of’daytime home-related factual’](image-url)
This technique produced clearly similar clusters within the 600 video dataset, for example, in this selection, you can clearly see items relating to women and also to drawing and art:

![Liza Donnelly: Drawing upon humor for change](image)

**Figure 15: Related items for Liza Donnelly video**

Our third example is an iPlayer version of N-screen. On any given day, there are about 1000 TV and radio programs available to UK viewers on iPlayer, the BBC’s on-demand service. This is an interesting dataset for us because of its high-quality metadata, including categories, formats, channel and people featured. We were curious as to whether we could generate interestingly similar program connections using only metadata. Our first approach was to try a Tanimoto similarity over the structured metadata, but the results were not particularly satisfactory – many programs had no similar items. We then tried tf-idf over the metadata descriptions. This seemed to pick up characteristics of the text rather than of the programs (for example repeated quirks in phrasing of the descriptions). The best approach we have tried (evaluated only informally) is tf-idf over a combination of metadata and the results of an entity-recognition technique.

We used the existing metadata from /programmes JSON format (for example [http://www.bbc.co.uk/programmes/b00k7pvx.json](http://www.bbc.co.uk/programmes/b00k7pvx.json) or [http://www.bbc.co.uk/programmes/b015ms3r.json](http://www.bbc.co.uk/programmes/b015ms3r.json)). As you can see from those examples, some have descriptions of people who are in the program, with mappings to dbpedia where available. We can get more of these by using a service to extract entities from the description text. For this we used Lupedia, which was developed in the NoTube project by Ontotext. We took this data coupled with the channel and the categories to produce a list of keywords, and then ran tf-idf over the top of that. The result can be variable:
Figure 16: Example of a not very good similarity match but in many cases, reasonably good

Figure 17: Example of a good selection of similar material and occasionally throws up an interesting surprise
8. Datasets

To test the pattern-based recommenders described in Section 3.2 we used the following datasets:

- **BBC programs as items**: A set of 12,000 BBC programs (via the open access of metadata in RDF from BBC /programmes). The names of programs and people are enriched with entities from the LD cloud, e.g. “Bill Cosby” is replaced with http://dbpedia.org/resource/Bill_Cosby, where additional machine-readable metadata of this person can be found.

- **user interests that are fetched from the social Web**: From Facebook and Twitter we fetched interests of 5 users, with their consent, using the NoTube Beancounter service. We selected 15 topics of interest, 8 people (e.g., actors, directors, “Rowan Atkinson”) and 7 programs (e.g. movies, “Zoolander” and “Burn After Reading”) that are related to five or more items by our recommender service.

- **LD as background knowledge**: As a background knowledge we used LinkedMDB, DBpedia and Freebase. In particular, we use the predicates that are mentioned earlier as well as owl:sameAs triples which link entities from the datasets that represent the same real-world objects. We also added a relatively small amount of triples that are required to compare the participant roles in the LinkedMDB dataset with those in the BBC EPG metadata. For example, triples with the predicate linkedmdb:actor express that a participant is featured in a movie with the role “actor”. However, this role “actor” is implicit in the predicate and cannot be directly compared as such with the BBC EPG metadata, where participant roles are explicit.

For the general analysis of LD to define general LD-patterns we have use the following three LD datasets:

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Jamendo</th>
<th>JPeel</th>
<th>LMDB</th>
</tr>
</thead>
<tbody>
<tr>
<td>nTriples</td>
<td>1,047,950</td>
<td>271,369</td>
<td>6,147,978</td>
</tr>
<tr>
<td>nProps</td>
<td>24</td>
<td>24</td>
<td>221</td>
</tr>
<tr>
<td>nTypes</td>
<td>11</td>
<td>9</td>
<td>53</td>
</tr>
</tbody>
</table>
Jamendo: is an online distributor of independent musical artists. The represented data focus on record authorship, release and distribution over internet channels. Being part of the DBTune service, its data representation relies on the Music Ontology and parts of FOAF, Dublin Core, Event, Timeline and Tags ontologies. The indie nature of its hosted artists, who are scarcely represented in other datasets, makes Jamendo a primary source for its content.

John Peel Sessions (JPeel): includes data related to live musical performances for the John Peel Show aired on BBC Radio One, and the resulting record releases. It is also a DBTune dataset, but being more event-focused, it mostly reuses a different portion of the Music Ontology vocabulary than Jamendo does.

LinkedMDB (LMDB28): is a triplified database of the film industry domain. It encompasses the entities involved with film production and release, plus additional metadata concerning ratings and events such as film festivals. The LinkedMDB ontology is unpublished and almost entirely in-house, with a few exceptions such as FOAF and Dublin Core terms.

We excluded general-purpose and cross-domain datasets, e.g. DBPedia and GeoNames, based on the already existing research experience on applying patterns to them. Examples are [41], which addresses the application of patterns to general-purpose datasets such as WordNet, and [42] which applies patterns to the Thesaurus of Geographic Names addressing geographical as well as art-related knowledge. In other words, we opted to experiment on a different type of resource in order to lay the basis - in our future work - for comparing our method and results with existing approaches.

9. Privacy
In NoTube we have identified several privacy requirements, based on the three use cases in the project. Below is a list of those requirements for User Interface and Interaction, both for the consumption of recommendations as well as in the User Profile (Beancounter):

9.1. Data ownership & portability:
- The UI must make clear that the user owns her own data.
- The user must be able to securely export her data at any time.
- It must be easy for the user to delete all her data at any time.
- The deleted data must be fully deleted from the Beancounter system.
- The user will have the option to have the deleted data emailed to her.
- The UI must make clear any policy which involves the user's data being used by third parties - e.g. to find out how many people watched a particular episode of a TV program. This use of an individual’s data would require ‘opt-in’ - we should assume that ticking a checkbox is not enough to obtain genuinely informed consent.

9.2. Sharing of data - UI principles:
- The default setting for all Beancounter-generated user data must be private until the user actively chooses to make their profile public.
- A user must be able to edit her profile before sharing it.
- A user can only ‘share’ her profile from the screen where she can see it: this is to protect people from sharing things inadvertently which they may prefer to keep private.
- A user can configure her Beancounter profile interests in a clear way that makes the privacy implications obvious.
- A user must be able to exclude some interests from her public profile (but retain them for recommendations).
- A user can start and stop sharing data at any time.
- A user can temporarily suspend sharing (equivalent of the "hide me" option in Fire Eagle).

28 http://www.linkedmdb.org/
Any updates or changes to the Profile are private by default. The user can be alerted to changes by email if she chooses to subscribe. If not, there is an ‘updated’ icon on the Profile screen and a facility to see the most recent changes. The user must approve any updates before they are shared.

A user can revert to a previous version of the Profile for sharing (with people or with applications) – therefore some sort of Profile Archive/history function is required.

Ideally, user should be alerted to privacy risks in the UI - e.g. "Are you sure you want to share this?". The exact mechanism for this is not yet known, but could include a status bar or traffic light system. Alternatively, the user could see who else can see a piece of information, or how many people can see it.

9.3. Sharing - user control & level of granularity:
The following requirements are speculative, and subject to further user experience research. In addition to keeping specific interests private, a user can also:

- specify individuals to share data with;
- specify individual applications to share data with (e.g. NoTube Recommender);
- keep specific events private - e.g. "I watched programme (or series) X";
- keep specific sources private - e.g. "Hide all data originating from my Twitter account".

We now discuss the issue of privacy and these requirements. If Beancounter supports sharing of personal data then we have to ask ourselves “What do we want the end result of sharing to be in the recommendations?”. We discussed having program recommendations with explainers such as “watch this because you’re a fan of David Attenborough, you’re interested in chimpanzees and your friends Libby and Yves watched it”, in which case all Libby and Yves have to share from their Beancounter is information about programs they have consumed – not their full Beancounter Profile of interests and contacts. Now, the main question is whether this has associated privacy risks in itself?

- Are there individual programs that Libby and Yves have watched that they might not want me to know about for various reasons?
- In which case do they need the option to keep specific ‘events’ private – e.g. ‘do not share the fact that I watched program (or series?) X’?
- Would this sharing of program information be reciprocal or can Libby see what I’ve watched without me having to know what she’s watched?
- Are there other scenarios for sharing which do involve sharing the full Profile (which is what the aggregation/enhancement privacy issues are about)? Might other people for example want to use your interests Profile to influence their own recommendations – because they believe they have similar tastes/interests to you? Is this something that Beancounter would support?
- If it is, can users make judgements about Beancounter privacy settings at the application level? Displaying the source of information would help here – e.g. “You watched Eastenders yesterday (Source: Twitter)”. This is something that would need to be tested and compared with a model whereby users can apply settings at the attention data item level (or types of items). A first step in sharing could be a Facebook widget - "Share your Profile with your Facebook friends using our Facebook application...".

Whilst the success of Beancounter-driven recommendations may rely, in part, on users sharing their Beancounter profiles with others (so that recommendations can be made on the basis of friends’ tastes as well as the user’s interests), it is important to give users maximum control over how their data is shared and with whom – so as to avoid any humiliation or embarrassments. However, this is easier said than done.

Firstly, recent BBC user research has shown that people are naturally quite guarded about sharing their TV viewing behaviours; they worry about how much time they can seen to be wasting by watching trashy TV (their ‘guilty pleasures’) which might not align with the public view of themselves they have presented online.
In line with Beancounter principles, the BBC research also found that people want sharing to be a conscious choice. However, it is very difficult to provide granular control of disclosure of personal information without imposing a heavy cognitive burden on the user. Facebook now has very extensive and precise privacy settings, but general consensus seems to be that they are very confusing and difficult to use.

This goes back to the previous point about how difficult is it for us to abstract privacy issues because we deal with them so effortlessly in everyday life, which might explain why many users never customise their privacy settings at all but just stick with the defaults (which, for social networking sites, are going to promote the sharing of data rather than keeping it private – without sharing there is no social network). This can be problematic when discretion is required: for example, earlier this year, the appointment of the new head of MI6 was exposed by his wife's Facebook page\(^\text{29}\). She had left all her Facebook privacy settings on their default settings, so all the information was available to any of the 200 million users in the open-access London network, as well as being searchable on Google. It seems that for Beancounter, the most appropriate default setting is ‘keep everything private until I choose to share it’.

When it comes to deciding what to share and with whom, it may be easier for users to focus on the ‘who’ i.e. the individuals they want to share with as the starting point, rather than focusing on the “what” i.e. the type of information to be shared (the latter seems to be the current user experience paradigm). A ‘who’ approach may align more closely with the way we think offline. For example, Lederer et al (2003) found that privacy preferences varied by ‘inquirer’ more than by situation – i.e. individuals were more likely to apply the same privacy preferences to the same person in different situations than to apply the same privacy preferences to different people in the same situation.

“For me, ‘who’ is all that matters. If I don’t trust the person with personal information, I wouldn’t want to give them information at any time. If I do trust the person, I’m willing to give out information freely”\(^\text{30}\)

However, there are exceptional circumstances where the situation may be more important: for example, you don’t want your partner to know your location because you’re organizing a surprise for him. In the context of Beancounter, it’s also possible that you might not want your partner, for example, to see what you’ve been watching, which is why granular sharing of the ‘what’ also has to be supported. In another interesting project, Lipford et al (2008) found that presenting an audience-oriented view of profile information significantly improved the understanding of privacy settings on Facebook, which may be why there is now an option to “See how a friend sees your profile” (you then type in the person’s name).

The Locaccino project\(^\text{30}\), a location-based friend-finding service for Facebook based on research at Carnegie Mellon University found that: “... Most of [the researcher's] test subjects started out reluctant to share their every move, even with friends. But users generally warmed to the system after they found the "hide my location" button for when they wanted to drop off the map”\(^\text{31}\). Likewise users of Fire Eagle can temporarily stop people from seeing their location using a “Hide me” option.

Experimenting with the best ways to support granular control in the Beancounter UI is a major area of future investigation for Notube. The aim is to design, like Locaccino, a very simple feature which is easy to understand and use, and which gives users immediate control without requiring them to enter into a complex decision-making process.

Several areas of further NoTube user experience research have been identified so far: including presentation of privacy policies for informed consent and design of simple and effective tools for complex granular control of data sharing. In addition, also proper legal advice is needed, although the

\(^{29}\) [http://www.guardian.co.uk/politics/2009/jul/05/mi6-facebook-sawers-wife-miliband](http://www.guardian.co.uk/politics/2009/jul/05/mi6-facebook-sawers-wife-miliband)

\(^{30}\) [http://locaccino.org](http://locaccino.org)

law has yet to catch-up with ever-evolving needs for privacy. In the meantime we need to focus on ways of balancing our desire for people to share their Beancounter data in order to influence the recommendations that their contacts receive (because it seems that people respond better to recommendations made by people they know), with our obligations to ensure that people actually understand what the potential costs to privacy might be.

10. Conclusion

In this deliverable, we have shown various aspects of achieving the main goal of the NoTube project - to help people find and choose what TV content to watch in a continuously growing amount of TV content and channels. We have experimented with novel semantics-based approaches for recommending TV programs. The core goal of such approaches is to allow for finding new connections between TV programs (e.g. connections between people involved in TV programs). Central to those approaches is using Linked Data as background knowledge in order to make such connections more specific and meaningful. Further, our goal is to integrate the semantics-based approaches in a complementary fashion to optimize the finding of recommendation results for any user and in any context. We have shown how to summarize Linked Data sets by treating them as sets of connected knowledge patterns, in order to identify their core knowledge components. We have experimented on three datasets from the LD cloud, and showed how to build prototypical queries for them even when the ontologies that model them are unknown. We have planned, in our future work, to compare ontologies explicitly published and used for a dataset with the knowledge architecture that arises from our analysis. Next to these semantic pattern-based approaches, the recommendation service also uses standard statistical techniques for collaborative filtering based on rating data. In reality, such TV data is very sparse, and expanding TV data like this can bring more programs from the long tail to the user's attention. Another aspect of the recommendation is matching more directly the concepts in the user profile, possible distilled from the user's activity data, with concepts related to programs, for example as basis for news-related recommendations.

Our ongoing and future work focuses on extending our strategy, in order to (i) demonstrate how by aligning emerging KPs of a dataset to general KPs improves interoperability across datasets, and detects incompatibility issues (ii) compare analysis data about different datasets, and (iii) improve user interaction in searches for relevant content. We have planned to improve the method by performing additional analysis on an extensive coverage of the multimedia domain, and subsequently evaluate the cross-domain portability of our approach.

11. Works Cited

23. Malaise, V., Hollink, L., Gazendam, L.: The interaction between automatic annotation and query expansion: A retrieval experiment on a large cultural heritage archive. In: Bloehdorn,