

Enhancing Content-based Recommendation with the Task Model of Classification

Yiwen Wang¹, Shenghui Wang², Natalia Stash¹,
Lora Aroyo^{1,2}, and Guus Schreiber²

¹ Eindhoven University of Technology, Computer Science
{y.wang,n.v.stash}@tue.nl

² VU University Amsterdam, Computer Science
{l.m.aroyo,schreiber}@cs.vu.nl, swang@few.vu.nl

Abstract. In this paper, we define reusable inference steps for content-based recommender systems based on semantically-enriched collections. We show an instantiation in the case of recommending artworks and concepts based on a museum domain ontology and a user profile consisting of rated artworks and rated concepts. The recommendation task is split into four inference steps: realization, classification by concepts, classification by instances, and retrieval. Our approach is evaluated on real user rating data. We compare the results with the standard content-based recommendation strategy in terms of accuracy and discuss the added values of providing serendipitous recommendations and supporting more complete explanations for recommended items.

1 Introduction

In recent years, the Semantic Web has put great effort on the reusability of knowledge. However, most work deals with reusable ontology and ontology patterns, there is hardly any work on reusable reasoning patterns [4]. Following the terminology defined by van Harmelen and ten Teije [4], we aim to identify reusable knowledge elements for content-based recommender systems based on semantically-enriched collections. As a first attempt, we show an instantiation in the domain of museums. We analyze our demonstrator³ (called the “CHIP Art Recommender”) and decompose the recommendation task into four inference steps: (i) realization (recommending concepts explicitly related to rated artworks via artwork features); (ii) classification by concepts (recommending concepts explicitly related to rated concepts via semantic relations); (iii) classification by instances (recommending concepts implicitly related to rated concepts using the method of instance-based ontology matching); and (iv) retrieval (recommending artworks based on both rated and recommended concepts).

³ <http://www.chip-project.org/demo/>

2 Task and Inference Steps

The CHIP Art Recommender stores the user profile in the form of both a set of rated artworks/instances and a set of rated concepts. Based on the user profile and the museum domain ontology, the system recommends both related artworks and related concepts via explicit and implicit relations.

Table 1. The task of content-based recommendation

Input:	a user profile characterized as both a set of instance $I_{profile}$ and a set of concepts $C_{profile}$
Knowledge:	an ontology $O = (T, I)$ consisting of a terminology T and an instance set I
Output:	<p>a set of related concepts $(C^i \cup C^j \cup C^k)$ with</p> <p>C^i: $\text{Recommend}(I_{profile}, O) = \{(i, \in, c^i) \mid \exists i: i \in I_{profile} \wedge i \in c^i\}$</p> <p>$C^j$: $\text{Recommend}(C_{profile}, T) = \{(c^j \sim c) \mid \exists c: c \in C_{profile} \wedge c^j \sim c\}$</p> <p>$C^k$: $\text{Recommend}(C_{profile}, O) = \{(c^k \simeq c) \mid \exists c: c \in C_{profile} \wedge c^k \simeq c \wedge i \in c \wedge i \in c^k\}$</p> <p>and a set of related instances I' with</p> <p>I': $\text{Recommend}(C_{profile}, C^i, C^j, C^k, O) = \{(i', \in, c') \mid c' \in (C_{profile} \cup C^i \cup C^j \cup C^k) \wedge i' \in c'\}$</p>

As described in Table 1, we use formal preliminaries to define the task of content-based recommendation: a terminology T is a set of concepts c organized in a hierarchy. Instance i is a member of such concepts c and this is described as (i, \in, c) where \in refers to the membership relation. An ontology O consists of a terminology T and a set of instances I . Sometimes we write (T, I) instead of O if we want to refer separately to the terminology and the instance set of the ontology. In our case, instances refer to artworks and each artwork is described with a number of concepts. Based on the semantically-enriched Rijksmuseum collection [6], we specify three different kinds of relations: (i) artwork feature, (ii) semantic relation, and (iii) implicit relation.

(i) Artwork feature is an explicit relation between an artwork and a concept, denoted as (i, \in, c) . For example, the artwork “The Night Watch” is related to the concept “Rembrandt van Rijn” via the artwork feature “*creator*”, the concept “Amsterdam” via the artwork feature “*creationSite*” and the concept “Militia” via the artwork feature “*subject*”.

(ii) Semantic relation is also an explicit relation, but it links two concepts, denoted as (c_i, \sim, c_j) . In our case, based on the semantically-enriched museum collections, there are not only domain-specific relations (e.g. *teacherOf*, *style*), but general relations (e.g. *broader/narrower*) as well [6].

(iii) Implicit relation connects two concepts that do not have a direct link between each other, denoted as (c_i, \simeq, c_j) . This relation is built based on common artworks these two concepts both describe, although there are no explicit/direct links between them.

To decompose the task of content-based recommendation, we identified four inference steps (see Fig. 1): (i) realization, (ii) classification by concepts, (iii) classification by instances, and (iv) retrieval.

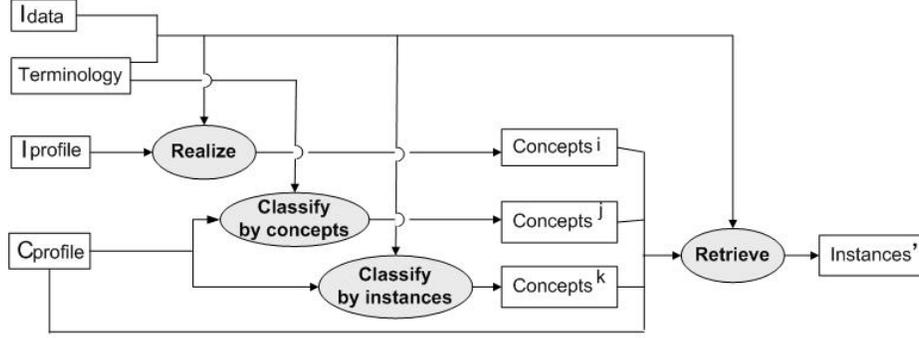


Fig. 1. Inference steps for the task of content-based recommendation

Realization is the task of finding a concept c that describe the given instances i .

- Definition: Find a concept c^i such that $O \vdash i \in c^i$
- Signature: $i \times O \mapsto c^i$

Classification by concepts is the task of finding a concept c^j which is directly linked to the given concept c through a semantic relation \sim in the hierarchy of terminology T .

- Definition: Find a related concept c^j through various semantic relations \sim (e.g. *broader*, *narrower*, *teacherOf*, *birthPlace*, etc.) in the terminology such that $T \vdash c \sim c^j$
- Signature: $c \times T \mapsto c^j$

Classification by instances is the task of finding a concept c^k which shares sufficient common instances with the given concept c using the instance-based ontology matching \simeq .

- Definition: Find a concept c^k through the instance-based ontology matching \simeq such that $O \vdash c \simeq c^k \wedge i \in c \wedge i \in c^k$
- Signature: $c \times O \mapsto c^k$

Retrieval is the inverse of realization: determining which instance i' belong to the related concept c' , where c' is a element of the unification of $C_{profile}$, C^i (Realization), C^j (Classification by concepts) and C^k (Classification by instances).

- Definition: Find an instance i' such that $i' \in c'$ where $c' \in (C_{profile} \cup C^i \cup C^j \cup C^k)$
- Signature: $c' \times O \mapsto i'$

Compared with the original definition of recommendation and its corresponding inference steps from van Harmelen and ten Teije [4], we extended the inference step of classification, which now consists of two components: classification by concepts and classification by instances. The main differences are: firstly, we applied much more different types of semantic relations [6] in the step of classification by concepts compared with the original classification which only uses the subsumption relation [4]; secondly, we proposed a new component “classification by instances”, which explores the implicit relations between concepts using the method of instance-based ontology matching from Issac et al. [2].

3 Semantic-Enhanced Recommendation Strategy

Suppose the user likes the artwork “The Little Street”, concepts “Rembrandt van Rijn” and “Venus”, Fig. 2 shows how the CHIP system recommends related concepts and artworks based on the user profile by taking four inference steps.

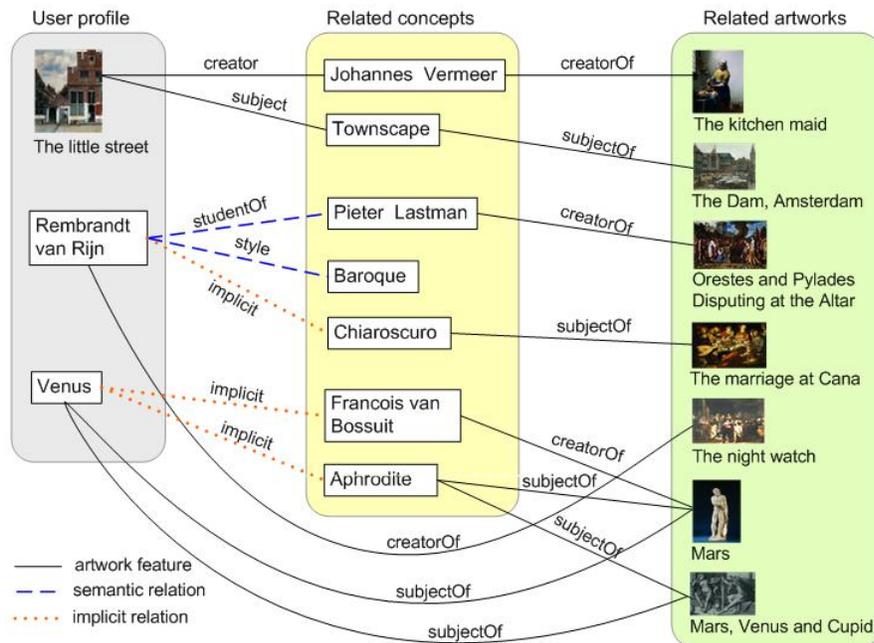


Fig. 2. Example of semantically-enhanced recommendations

• **Realization:** Based on the artwork “The Little Street”, it recommends the concept “Johannes Vermeer” via the artwork feature *creator* and the concept “Townscape” via the artwork feature *subject*.

- **Classification by concepts:** Based on the concept “Rembrandt van Rijn”, it recommends the concept “Pieter Lastman” via the semantic relation *studentOf* and the concept “Baroque” via the semantic relation *style*.

- **Classification by instances:** Based on the concept “Rembrandt van Rijn”, it recommends the concept “Chiaroscuro” because they share sufficient (by setting the threshold) common artworks. Based on the concept “Venus”, it recommends concepts “Francois van Bossuit” and “Aphrodite” also because of the sufficient common artworks they describe.

- **Retrieval:** Based on three sets of concepts: (i) rated concepts (“Rembrandt van Rijn” and “Venus”); (ii) explicitly related concepts via artwork features and semantic relations (“Johannes Vermeer”, “Townscape”, “Pieter Lastman” and “Baroque”); and (iii) implicitly related concepts (“Chiaroscuro”, “Francois van Bossuit” and “Aphrodite”), it recommends artworks “The Kitchen Maid”, “The Dam, Amsterdam”, “Orestes and Pylades Disputing at the Altar”, “The Marriage at Cana”, “The Night Watch”, “Mars” and “Mars, Venus and Cupid” via artwork features *creatorOf* and *subjectOf*.

3.1 Computing the Explicit Value for the Steps of Realization and Classification by Concepts

In a previous user study [6], we explored the use of various explicit relations between artworks and concepts for recommendations. These relations include: (i) artwork features between an artwork and concepts (e.g. *creator*); and (ii) semantic relations between two concepts within one vocabulary (e.g. *broader*) and across two different vocabularies (e.g. *style*).

Table 2. Weights of explicit relations

Relation	creator	creation Site	subject Of	style Of	birth Place	death Place	teacher Of	aat Broader	tgn Broader	ic Broader
Weight	0.67	0.35	0.50	0.63	0.32	0.26	0.43	0.53	0.22	0.50
Inverse Relation	creator Of	creation SiteOf	subject Of	style Of	birth PlaceOf	death PlaceOf	student Of	aat Narrower	tgn Narrower	ic Narrower
Weight	0.68	0.31	0.54	0.61	0.28	0.21	0.44	0.55	0.16	0.52

Using the existing user ratings collected from this study, we investigated the preliminary weights $W_{(r)}$ (see Table 2) for each explicit relation $R_{(i,j)}$, which is either an artwork feature between an artwork i and a concept j or a semantic relation between two concepts (i and j). For example, the relation between artwork “The Little Street” and concept “Johannes Vermeer” is *creator*, denoted as $R_{(TheLittleStreet,JohannesVermeer)} = creator$. From Table 2, we know that the weight of this relation $W_{(creator)}$ is 0.67. In the formulas below we write $W_{(i,j)}$ instead of $R_{(i,j)}$ and $W_{(r)}$.

Considering that a rated item (either an artwork or a concept) could be linked to multiple items via various explicit relations, we need to normalize

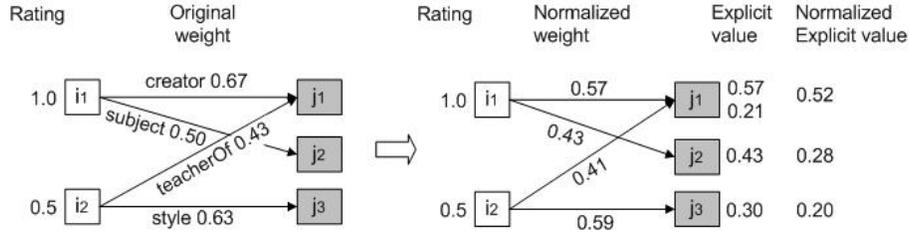


Fig. 3. Example of calculating the normalized explicit value

the weight(s) for each related item. As shown in Fig. 3, the rated item i_1 is linked to items j_1 and j_2 . The relation between i_1 and j_1 is *creator* and the corresponding weight of *creator* is denoted as $W_{(i_1, j_1)}$. From Table. 2, we know that $W_{(i_1, j_1)}$ (*creator*) is 0.67, $W_{(i_1, j_2)}$ (*subject*) is 0.50, $W_{(i_2, j_1)}$ (*teacherOf*) is 0.43, and $W_{(i_2, j_3)}$ (*style*) is 0.63.

To normalize the weights, Formula 1 is applied. For example, based on i_1 , the normalized weight of j_1 : $NW_{(i_1, j_1)} = \frac{0.67}{0.67+0.50} = 0.57$ and the the normalized weight of j_2 : $NW_{(i_1, j_2)} = \frac{0.50}{0.67+0.50} = 0.43$. In this way, we could calculate that based on i_2 , normalized weight of j_1 : $NW_{(i_2, j_1)} = \frac{0.43}{0.43+0.63} = 0.41$ and the normalized weight of j_3 : $NW_{(i_2, j_3)} = \frac{0.63}{0.43+0.63} = 0.59$.

Formula 1: Normalized weight	Formula 2: Explicit value	Formula 3: Normalized explicit value
$NW_{(i,j)} = \frac{W_{(i,j)}}{\sum_{j=1}^J W_{(i,j)}}$	$Exp_{(i,j)} = NW_{(i,j)} \times R_{(i)}$	$NExp_{(j)} = \frac{\sum_{i=1}^I Exp_{(i,j)}}{\sum_{i=1}^I \sum_{j=1}^J Exp_{(i,j)}}$

Based on the normalized weights and user ratings, the next step is to compute the semantic value, see Formula 2. Based on i_1 , the semantic values of j_1 and j_2 are: $Exp_{(i_1, j_1)} = 0.57 * 1.0 = 0.57$, and $Exp_{(i_1, j_2)} = 0.43 * 1.0 = 0.43$. Based on i_2 , $Exp_{(i_2, j_1)} = 0.41 * 0.5 = 0.21$, and $Exp_{(i_2, j_3)} = 0.59 * 0.5 = 0.30$.

Finally, we also need to normalize these semantic values for each related item, see Formula 3. $NExp_{j_1} = \frac{0.57+0.21}{0.57+0.21+0.43+0.30} = 0.52$; $NExp_{j_2} = \frac{0.43}{0.57+0.21+0.43+0.30} = 0.28$; and $NExp_{j_3} = \frac{0.30}{0.57+0.21+0.43+0.30} = 0.20$.

3.2 Computing the Implicit Value for the Step of Classification by Instances

Sometimes there is no explicit relations between two concepts, however, they could be actually very similar or close to each other via some implicit relations. For example (see Fig. 2), “Rembrandt van Rijn” is famous for his technique

using strong contrast of light and dark shading, which in Italian corresponds to “Chiaroscuro”; “Francois van Bossuit” often took “Venus” as a subject to paint; and “Venus” in Roman refers to “Aphrodite” in Greek. Compared with the “obvious recommendations” via explicit relations, these implicitly related concepts might be surprisingly new/unknown to users. The main challenge is to define how close these two concepts are in the collection.

To address this issue, Issaac et al. [2] propose a method of instance-based ontology matching. The basic idea is that the more significant the overlap of artworks of two concepts is, the closer these two concepts are, and the level of significance is calculated by the corrected Jaccard measure, see Formula 4. In the formula, the set of instances described by a concept c is called the extension of c and abbreviate by C^i . The $JCcorr(C_1, C_2)$ measures the fraction of the refinement (by choosing the factor of 0.8) of instances described by both concepts C_1 and C_2 relative to the set of instances described by either one of the concepts [2].

$$JCcorr(C_1, C_2) = \frac{\sqrt{|C_1^i \cap C_2^i| \times (|C_1^i \cap C_2^i| - 0.8)}}{|C_1^i \cup C_2^i|} \quad (\text{Formula 4: Corrected Jaccard measure})$$

Adopting this method, we calculated the Corrected Jaccard values for all pairs of concepts in the collection. In general, the higher the Corrected Jaccard value is, the more common artworks these two concepts described. Below we give a brief look at the Corrected Jaccard values for some pairs of concepts:

- 0.96 (Sculptural studies – Terracotta models)
- 0.91 (unknown lacquerer – Lacquerware)
- 0.85 (Hermes – Mercury)
- 0.75 (Food and other objects – Still lifes with food)
- 0.63 (Militias – Militia paintings)
- 0.50 (Hinduism – Hindu deities)
- 0.40 (Still-life painting – Food and other objects)
- 0.30 (Drinking games – Sport and Games)
- 0.20 (Cupid – Love and Sex)
- 0.15 (Polychromy – Golden Legend)
- 0.10 (Rendering of texture – Woman)

There are in total 24249 pairs of concepts and the range of the Corrected Jaccard value is between 0 and 1. Looking at these values and checking the corresponding number of artworks the pair of concepts describe in common, we set 0.20 as a preliminary threshold, which might needs more refinement in the future. An example for the threshold 0.20 is “Cupid” and “Love and sex”, which describe 8 artworks in common out of 40 artworks that are described by either one of these two concepts. In comparison, the Corrected Jaccard value between “Rendering of texture” and “Woman” is 0.10 and they describe 4 artworks in common out of 41 artworks.

After getting the Corrected Jaccard values for all concept pairs, we follow the same steps (Formula 1, 2 and 3) as the calculation of the explicit semantic value in Section 3.1. The only difference is that we use the Corrected Jaccard value to replace the original weight between two concepts and then normalize the Corrected Jaccard value in Formula 1. In the end, we will get a normalized implicit value $NImp_{(j)}$ for each implicitly related concept j .

3.3 Combining the Explicit and Implicit Values for the Step of Retrieval

Considering a related concept j could be linked to rated items via not only explicit relations but also implicit relations, we need to combine values from these two parts in order to get a final prediction $PreC_{(j)}$ for recommendation. Inspired by the work from Mobasher et. al [3], we set a parameter α to combine these two parts, see Formula 5. This combination parameter α measures the strength of the explicit and implicit components with respect to the current context. Taking two extreme examples: When α is 1, the system recommends items purely based on explicit relations and this will work well if the collection is well structured with rich semantic relations. When α is 0, it recommends items purely based on implicit relations which is suitable for recommender systems working on databases without semantic structures between concepts. Ideally, the parameter α could be manually set by the user, or dynamically adapted by the system, which enables the flexibility of the recommendation algorithm.

$$PreC_{(j)} = \alpha \times NExp_{(j)} + (1 - \alpha) \times NImp_{(j)}$$

(Formula 5: Prediction for related concepts)

After collecting related concepts via both explicit and implicit relations, the system retrieves related artworks based on these related concepts. Since there are only explicit relations, which are artwork features between concepts and artworks, we only need to compute the normalized semantic value for related artworks, which is explained in details in Formula 3.

4 Evaluation and Discussion

In the evaluation, we use the existing user ratings collected from the previous study [6]. There were 48 users that participated in this study. They used the CHIP Art Recommender to browse the Rijksmuseum collection, which contains 729 artworks and 4320 art concepts. Each user rated 53 items (artworks and concepts) on average. We evaluate the recommendation accuracy and discuss the added values of providing serendipitous recommendations and explanations for recommended items.

To measure the recommendation accuracy, we compute the standard Mean Absolute Error (MAE) by Leave-one-out cross validation [1]. MAE measures the average absolute deviation between ratings and predictions. Although there are a number of variables influencing the MAE (e.g. the parameter α , the weights for explicit relations and the threshold for the Corrected Jaccard value), in this evaluation, we only look at the impact of α on MAE in order to get a first insight and we leave the experimentation with other variables to future work.

In order to see whether the semantic-enhanced content-based recommendation (SE-CBR) strategy in general improves or hamper the accuracy, we also measure the MAE for the standard content-based recommendation (CBR) strategy, which was applied in the previous version of the system [5]. The standard

CBR takes the inference steps of realization and retrieval, but no classification by concepts and instances, which means that based on user rated items, standard CBR only recommends items via artwork features.

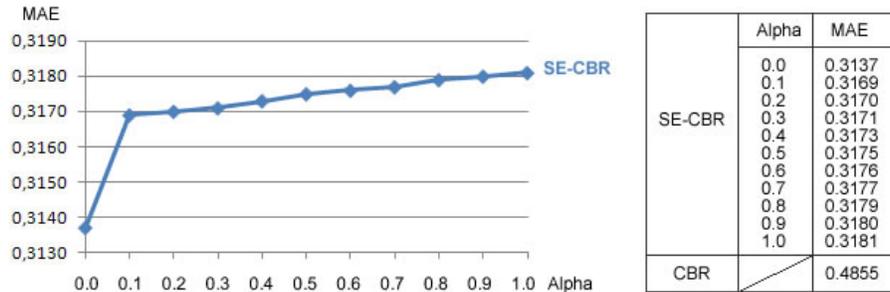


Fig. 4. MAE for SE-CBR and CBR

Note that ratings in our system are based on a 5-star scale, which refers to -1, -0.5, 0, 0.5, 1. Thus the maximum possible value for MAE is 2 and the minimum value is 0. The lower MAE represent the higher recommendation accuracy. In Fig.4, we observe that: (i) Compared with CBR (MAE is 0.4855), SE-CBR reaches a much lower MAE, which is in the range of 0.3137 (α is 0) and 0.3181 (α is 1). It shows that although recommending more items, SE-CBR does not sacrifice the recommendation accuracy, surprisingly, it even improves the accuracy compared with CBR. (ii) The impact of α on MAE for SE-CBR is not significant, with a slight increase from 0.3137 (α is 0) to 0.3181 (α is 1). The reason could be that we set a very high threshold (0.20) for the Corrected Jaccard value when selecting implicitly related items. Among all 24249 pairs of concepts in the collection, only 4% (1175 pairs) has the Corrected Jaccard value above 0.20 and most of these pairs are either synonyms or very similar to each other, e.g. “Unknown lacquerer”-“Lacquerware” and “Food and other objects”-“Still lifes with food”. The high similarity ensures a high accuracy for implicit recommendations. When α is 0, it only recommends implicitly related concepts which are kind of synonyms in our case and thus it reaches the lowest MAE value of 0.3137. Considering the majority (75%: 18186 concept pairs) has the Corrected Jaccard values between 0.01 and 0.10, if we set a threshold in a lower range, it will bring a lot of noisy recommendations, which might significantly decrease the recommendation accuracy. Besides the threshold for the Corrected Jaccard value, there are a number of parameters (e.g. weights for explicit relations) that influence the accuracy. We plan to try a machine learning based approach instead of the manual turning in follow up work.

As Herlocker et al. [1] argued, accuracy alone is not sufficient for selecting a good recommendation algorithm. A serendipitous recommendation helps a user find a surprising and new/unknown item that he/she might not have otherwise

discovered. Besides, explanations of why an item was recommended also helps users gain confidence in the system’s recommendations. As illustrated in Fig. 2, if a user likes the famous Dutch painter “Rembrandt van Rijn”, the standard CBR could only recommend the artwork “The Night Watch” via the artwork feature *creatorOf*. In comparison, the SE-CBR could recommend more items besides “The Night Watch”: (i) by taking the step of classification by concepts, it recommends concepts “Baroque” (*style*) and “Pieter Lastman” (*studentOf*) based on the semantic relations between concepts; (ii) by taking the step of classification by instances, it recommends an implicitly related concept “Chiaroscuro” based on instance ontology matching; (iii) by taking the step of realization, it recommends artworks “The Marriage at Cana” and “Orestes and Pylades Disputing at the Altar” based on all related concepts. For each recommended item, the system provides the explanation of “Why recommend”, which automatically derives relations between the user’s rated items and recommended items from the domain ontology. In such a way, the user could receive not only more recommended items, but also more complete explanations, which could help them better understand the recommendations. A further user study is needed to evaluate the aspects of serendipity and explanations.

In this work, our intention was to identify reusable knowledge elements for content-based recommender systems based on semantically-enriched collections. We demonstrated our approach in the domain of museum art collections. In future work, we plan to test this approach for different applications and ontologies.

5 Acknowledgements

We greatly appreciate the contribution of Annette ten Teije from VU University Amsterdam for providing the inspiration to our work and the discussion of the results.

References

1. Jonathan L. Herlocker, Joseph A. Konstan, Loren G. Terveen, John, and T. Riedl. Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems*, 22:5–53, 2004.
2. Antoine Isaac, Lourens Van der Meij, Stefan Schlobach, and Shenghui Wang. An empirical study of instance-based ontology matching. In *6th International and 2nd Asian Semantic Web Conference (ISWC ’07 + ASWC ’07)*, pages 252–266, 2007.
3. Bamshad Mobasher, Xin Jin, and Yanzan Zhou. Semantically enhanced collaborative filtering. In *Web Mining: From Web to Semantic Web*, 3209, 2004.
4. Frank van Harmelen, Annette ten Teije, and Holger Wache. Knowledge engineering rediscovered: towards reasoning patterns for the semantic web. In *5th International Conference on Knowledge Capture (K-CAP ’09)*, pages 81–88, 2009.
5. Yiwen Wang, Natalia Stash, Lora Aroyo, Peter Gorgels, Lloyd Rutledge, and Guus Schreiber. Recommendations based on semantically-enriched museum collections. *Journal of Web Semantics*, 6(4):43–58, 2008.
6. Yiwen Wang, Natalia Stash, Lora Aroyo, Laura Hollink, and Guus Schreiber. Semantic relations for content-based recommendations. In *5th International Conference on Knowledge Capture (K-CAP ’09)*, pages 209–210, 2009.