Property-based Semantic Similarity and Relatedness for Improving Recommendation Accuracy and Diversity

Silvia Likavec, Computer Science Department, University of Torino, Torino, Italy

Francesco Osborne, KMi, The Open University, Milton Keyes, UK

Federica Cena, Computer Science Department, University of Torino, Torino, Italy

ABSTRACT

The authors introduce new measures of semantic similarity and relatedness for ontological concepts, based on the properties associated to them. They consider two concepts similar if, for some properties they have in common, they also have the same values assigned to these properties. On the other hand, the authors consider two concepts related if they have the same values assigned to different properties. These measures are used in the propagation of user interest values in ontology-based user models to other similar or related concepts in the domain. The authors tested their algorithm in event recommendation domain and in recipe domain and showed that property-based propagation based on similarity outperforms the standard edge-based propagation. Adding relatedness as a criterion for propagation improves diversity without sacrificing accuracy. In addition, assigning a certain relevance to each property improves the accuracy of recommendation. Finally, the property-based spreading activation is effective for cross-domain recommendation.

Keywords: *Intelligent System, Ontology, Properties, Relatedness, Similarity, User Interest Propagation, User Model*

1 INTRODUCTION

Semantic similarity and semantic relatedness are two important concepts, which find their application in many areas, from Natural Language Processing (NLP) and Information Retrieval to Semantic Web. *Semantic similarity* usually accounts for common features in concept definitions, whereas *semantic relatedness* takes into account any functional or lexical relation between concepts as well (Hirst & St-Onge, 1998). Usually, semantic similarity considers only subsumption relation (mammal-cat), whereas semantic relatedness considers other types of relations, such as

DOI: 10.4018/IJSWIS.2015100101

Copyright © 2015, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

hyponymy/hypernymy, meronomy/holonymy, etc., as well as any other kind of functional relationship or frequent association (cat-milk). Hence, semantic similarity can be seen as a special case of semantic relatedness (Resnik, 1995) and is much more difficult to compute.

In this work we look at the similarity and relatedness of concepts in domain ontologies, where concepts are distinguished by their position in a taxonomy derived from the ontology and by the properties associated to them. Ontologies (Gruber, 1993; Guarino & Poli, 1995) are widely adopted to represent and model domain knowledge in web applications, and lot of attention is paid to the development of new ontology languages and associated reasoning mechanisms. Ontological representation of domain objects allows for representing both similarities and differences among domain objects as well as generic objects and very specific ones, allowing to define uniform classification criteria. Ontologies also include relations among objects and make it possible to derive implicit information from explicitly represented knowledge.

Drawing inspiration from Tversky's work on *Features of Similarity* (Tversky, 1977), our working hypothesis on similarity and relatedness among ontological objects, where objects include concepts and their instances, is the following: two objects are *similar* if they both are defined having the same properties with the same values and two objects are *related* if they are defined having the same values for different properties. For example, in the event domain, Cooking_course and Dinner are similar concepts since both are defined having the same value Restaurant for the property has_venue_type. On the other hand, Football_game has the value Sport for the property has_main_argument, whereas Stadium has the same value Sport for a different property is_venue_for. The same happens for instances. For example, the two movies Hair and Amadeus are similar since they have the same value Forman for the property has_director, whereas the movies Desperately_Seeking_Susan and MTV_Video_Music_Awards are related since they have the same value Madonna for the properties has_main_actor and has_singer, respectively. Hence, the related objects are somehow connected, even when they are not similar.

We apply this semantic similarity and relatedness measure to the propagation of user interest values in ontology-based user models. A user model (Kobsa et al., 2001) is a knowledge structure which maintains users' features (such as demographic features) and users' attitudes (such as interest and knowledge) in the main domain objects. In an *ontology-based user model*, the user interest values are assigned to domain concepts and items as an overlay over a domain ontology (Brusilovsky, 2007). In these kinds of user models it is possible to exploit the ontological structure in order to propagate user values over ontology objects, starting from a small number of initial objects to other similar and related objects, and to incrementally update the user model (Cena et al., 2012; IJntema et al., 2010; Sieg et al., 2007).

This propagation of user interests can be seen as a special case of spreading activation theory (Salton and Buckley, 1988), since starting from a user interest in one domain object, it is possible to propagate this interest to other somehow similar or related objects. For example, knowing that a user is interested in the pop singer Madonna, we can reasonably infer user interest also in Pop_music and Music in general (both more general concepts), in Sting (a similar instance) but also in the movie Desperately_Seeking_Susan where Madonna plays the main role (a related instance). This is particularly useful to solve the cold start problem (Salton and McGill, 1984), which happens at the beginning of the interaction when a system does not have enough information about users.

Using relatedness in addition to similarity when determining to which domain objects to propagate the initial user interest makes it possible to provide the users with the results that can be very different from each other (and would be neglected by approaches that rely exclusively on semantic similarity), hence helping to solve the diversity problem in recommendation. The diversity problem (O'Sullivan et al., 2004) occurs when recommendation results, although similar

Copyright © 2015, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

to the initial object, are also very similar to each other, thus lacking diversity and not providing the user with satisfying alternatives. In this sense, relatedness complements similarity, since it brings more diversity into recommendation process and provides the user with more serendipitous recommendations. However, providing recommendations that are very different from the initial user interest can decrease the accuracy of recommendation results.

The motivation for the present research stems from the following goals:

1. To improve the *accuracy* of recommendation results obtained by propagation of user interests in ontology-based user models;
2. To improve the *diversity* of recommendation results obtained by propagation of user interests in ontology-based user models, without decreasing the accuracy too much.

In order to reach these goals, we propose to extend the user interests propagation method presented in (Cena et al., 2012) with (i) the *relevance of properties* to improve accuracy and (ii) the *relatedness among the domain objects*, in addition to similarity, to improve diversity.

We tested our method in two domains, namely event recommendation and recipes recommendation, in order to verify the following hypotheses (H4 was not tested in the second evaluation):

H1. Property-based user interest propagation outperforms the classic edge-based user interest propagation described in the state of the art (Cena et al., 2011, 2013; Middleton et al., 2004; Sieg et al., 2007;);

H2. Assigning a certain weight or relevance to each property brings a significative improvement to the accuracy of recommendation derived from the propagation;

H3. Property-based similarity offers good results w.r.t. diversity. In addition, considering relatedness in propagation yields a good increment to recommendation diversity, paying a relatively low cost in accuracy;

H4. Property-based user interest propagation is particularly effective when propagating user interest values across different sub-domains to provide cross-domain recommendation.

Note that our approach using only similarity among objects was tested in a different domain (gastronomy) in (Cena et al., 2012). Hence, with respect to (Cena et al., 2012), the *main contributions* of this work are the following:

1. In addition to property restrictions on classes, cardinality restrictions are taken into account;
2. Relatedness among objects is fully exploited to improve diversity, as opposed to the approach in (Cena et al., 2012) where relatedness was only briefly mentioned;
3. A novel approach for calculating the relevance of properties was introduced, to improve the recommendation accuracy;
4. The approach for user interest propagation was tested in two additional domains (events recommendation and recipes).

The rest of the paper is organised as follows. In Section 2, we turn to ontologies for knowledge representation and give a brief description of the treatment of properties in OWL. In Section 3 we provide the details of how to calculate the similarity and relatedness between objects in the domain ontology, followed by the description of how to detect the relevance of properties in Section 4. In Section 5 we describe our user modelling approach and the specific process of propagation of user interests in an ontology, based on properties of domain objects. In Section

Copyright © 2015, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

6 we report the evaluation results for the domain of event recommendation and for the domain of recipes recommendation. We present the most relevant related work in the field in Section 7. Finally, we conclude with some directions for future work in Section 8.

2. BACKGROUND: OWL ONTOLOGIES FOR KNOWLEDGE REPRESENTATION

In many different areas, ontologies proved to be indispensable tools for representing domain knowledge semantically (Antoniou & van Harmelen, 2008; Allemang & Hendler, 2008), since the powerful formalisms which they are based on enable explicit specification of domain elements and their properties, hierarchical organization of domain elements, as well as precise description of the relationships between them and usage of rigorous reasoning mechanisms. An ontology can be seen as a formal, explicit specification of a shared conceptualization (Gruber, 1993). One standard formalism for representing ontologies is OWL (http://www.w3.org/TR/owl-ref).

2.1. Properties in OWL

In ontologies expressed in OWL the features of domain elements are expressed using properties. There are two kinds of properties in OWL:

1. *Object properties* describing relations among individuals and
2. *Data type properties* providing relations among individuals and data type values.

Object properties and datatype properties are defined as instances of the built-in OWL classes

```
owl:ObjectProperty
owl:DatatypeProperty,
```

To define the characteristics of a property the *property axiom* is used, most commonly defining its domain and range. rdfs:domain links a property to a class description, whereas rdfs:range links a property to either a class description or a data range. For example:

```
<owl:ObjectProperty rdf:ID="has_location">
 <rdfs:domain rdf:resource="#Event"/>
 <rdfs:range rdf:resource="#Location"/>
</owl:ObjectProperty>
```

Equivalent properties are defined with owl:equivalentProperty.

Apart from explicitly defining properties for the classes, properties can be used to define classes with property restrictions, as we will see in the following section.

2.2. Defining Classes with Property Restrictions

Properties in OWL are used for defining classes with property restrictions. This is done by means of local anonymous classes, grouping together all the individuals which satisfy certain restrictions on certain properties. In OWL, two kinds of property restrictions can be distinguished:

1. *Value constraints* concerning constraints on the range of the property when applied to a particular class description and
2. *Cardinality constraints*, imposing constraints on the number of values a property can take, in the context of a particular class description.

Copyright © 2015, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

Value constraints can be defined in the following three ways:

1. owl:allValuesFrom defines a class for which all the values of the given property are either members of the specified class or data values within the specified data range. Not having any values for the given property is possible. For example:

```
<owl:Restriction>
 <owl:onProperty rdf:resource="#has_main_actor" />
 <owl:allValuesFrom rdf:resource="#Actor" />
</owl:Restriction>
```

   defines an anonymous OWL class of all the individuals for which the has_main_actor property only has values of the class Actor. Note that the property does not always have the values of this class; rather it is true for individuals belonging to the class extension of the anonymous restriction class.
2. owl:someValuesFrom defines a class for which at least one of the values of the given property is either a member of the specified class or a data value within the specified data range. At least one value for the given property must exist. Other values for the given property might exist. For example:

```
<owl:Restriction>
 <owl:onProperty rdf:resource="#has_main_actor" />
 <owl:someValuesFrom rdf:resource="#Singer" />
</owl:Restriction>
```

   defines an anonymous OWL class of all the individuals which have at least one Singer as the value of the property has_main_actor.
3. owl:hasValue defines a class for which the specified property has at least one value semantically equal to the specified value, which can be either an individual or a data value. "Semantically equal" datatypes are the ones which have the lexical representation of the literals mapped to the same value. "Semantically equal" individuals either have the same URI reference or are defined as being the same individuals with owl:sameAs. For example:

```
<owl:Restriction>
 <owl:onProperty rdf:resource="#has_main_actor" />
 <owl:hasValue rdf:resource="#Madonna" />
</owl:Restriction>
```

On the other hand, cardinality constraints can be defined in the following three ways:

1. owl:maxCardinality defines a class which has at most *max* semantically distinct values for the specified property (where *max* is the value of the cardinality constraint). For example:

```
<owl:Restriction>
<owl:onProperty rdf:resource="#has_number_of_workshops" />
<owl:maxCardinality rdf:datatype="&xsd;nonNegativeInteger">5 </
```

Copyright © 2015, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

```
owl:maxCardinality>
</owl:Restriction>
```

describes an anonymous OWL class of individuals that have at most five workshops (for example conferences affiliated with 1 − 4 workshops).
2. owl:minCardinality is defined analogously to owl:maxCardinality, and it defines a class which has at least *min* semantically distinct values for the specified property (where *min* is the value of the cardinality constraint).
3. owl:cardinality is defined analogously to owl:maxCardinality or owl:minCardinality, defining the class which has exactly *m* semantically distinct values (where *m* is the value of the cardinality constraint). It is actually a redundant concept, since it can be defined by using the combination of owl:maxCardinality and owl:minCardinality.

So we have seen how to define properties for domain concepts and how to define class restrictions using properties in order to describe the concepts in the ontology. We can consider each of the concepts in our ontology to have certain properties defined for it. These properties further describe the concepts in the ontology and can be used to calculate their mutual similarity.

2.3. Instances and their Properties

Instances in the ontology are defined using individual axioms describing their class memberships, individual identities and property values. Instances basically inherit their properties from the classes they are instances of. They are related to the classes they belong to directly with the rdfs:type relation. Therefore, in OWL instances have a specific value associated to each property. For example:

```
<Concert rdf:ID=″Mozart_for_all″>
 <has_concert_genre rdf:resource=″#classical″/>
 <has_composer rdf:resource=″#Mozart″/>
 <has_director rdf:resource=″#Simon_Rattle″/>
 <has_location rdf:resource=″#Torino″/>
 <has_venue rdf:resource=″#″Auditorium″/>
</Concert>
```

3. PROPERTY-BASED SIMILARITY AND RELATEDNESS OF DOMAIN OBJECTS

We begin this section with an example which should help illustrate our idea. In an ontology describing events, certain concepts can have these properties defined: has_venue, has_main_argument, has_price.

Consider the following events (instances) from this ontology: Picasso_Exhibition, Exhibition_from_Botticelli_to_Matisse and Human_Body_Exhibition. All of them have Art as the main argument, the first two have Gallery as their venue, whereas the third one has Stadium as its venue. Picasso_Exhibition costs 10 euros, Exhibition_from_Botticelli_to_Matisse costs 7 euros and Human_Body_Exhibition costs 12 euros. If we simply count the properties and their corresponding values which certain events have in common, we see that Picasso_Exhibition is more similar to Exhibition_from_Botticelli_to_Matisse than to Human_Body_Exhibition, since Picasso_Exhibition and Exhibition_from_Botticelli_to_Matisse have two properties in common

Copyright © 2015, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

with the same values, whereas Picasso_Exhibition and Human_Body_Exhibition have only one property in common with the same value. But, it can happen that for a certain property, two values are given. For example, for Human_Body_Exhibition, the property has tag has three values: Nature, Body and Science.

When calculating the property-based similarity of two domain objects O_1 and O_2 we want to account for their common and distinctive features. This is possible by using Tversky's feature-based model of similarity (Tversky, 1977), where similarity between objects is a function of both kinds of features of the objects in question:

$$sim_T(O_1,O_2)=\frac{\alpha(\psi(O_1)\cap\psi(O_2))}{\beta(\psi(O_1)\setminus\psi(O_2))+\gamma(\psi(O_2)\setminus\psi(O_1))+\alpha(\psi(O_1)\cap\psi(O_2))} \tag{1}$$

In the above formula $\psi(O)$ is the function describing all the relevant features of the object O, and $\alpha,\beta,\gamma\in\square$ are constants which permit different treatment of the various components. When $\alpha=1$ common features of the two compared objects are given maximal importance and for $\beta=\gamma$ non-directional similarity measure is obtained. We will set $\alpha=\beta=\gamma$. We will be using the following notation:

- *Common features of* O_1 and O_2: $cf(O_1,O_2)=\psi(O_1)\cap\psi(O_2)$,
- *Distinctive features of* O_1: $df(O_1)=\psi(O_1)\setminus\psi(O_2)$ and
- *Distinctive features of* O_2: $df(O_2)=\psi(O_2)\setminus\psi(O_1)$.

With this notation, taking $\alpha=\beta=\gamma=1$, the formula (1) becomes:

$$sim_T(O_1,O_2)=\frac{cf(O_1,O_2)}{df(O_1)+df(O_2)+cf(O_1,O_2)} \tag{2}$$

In order to calculate these values for domain objects O_1 and O_2, we need to calculate for each property *p,* how much it contributes to common features of O_1 and O_2, distinctive features of O_1 and distinctive features of O_2, respectdively. We denote these values by cf_p, df_p^1 and df_p^2, and calculate them depending on how different properties are defined for O_1 and O_2. We consider equal the properties defined with owl:EquivalentProperty.

3.1. Similarity among Classes Defined as Restrictions

When the classes are defined as property restrictions, it is obvious that the values for the property used to define them are limited. Depending on the size of the set of classes which can provide possible values for the given property, the effective values for common and distinctive features would differ. For example, in the case of restrictions defined with owl:allValuesFrom, all the values of the given property are members of the specified class (hence, also of its subclasses), in the case of restrictions defined with owl:someValuesFrom at least one of the values of the given property is a member of the specified class but other values for the given property might

Copyright © 2015, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

exist and in the case of restrictions defined with owl:hasValue the specified property has at least one value semantically equal to the specified value.

In the detailed technical description below, we first consider three kinds of value restriction declarations:

- owl:allValuesFrom;
- owl:someValuesFrom;
- owl:hasValue.

Depending on how the restrictions on properties are defined for O_1 and O_2, we distinguish the following six cases:

1. The property *p* is defined with owl:allValuesFrom for both O_1 and O_2. Let the property *p* be defined for, O_1 with

`<owl:allValuesFrom rdf:resource="#`A_1`">`
and for O_2 with
`<owl:allValuesFrom rdf:resource="#`A_2`">.`

Let a_1 (resp. a_2) be the number of subclasses of A_1 (resp. A_2). If A_1 and A_2 are equal classes or declared equivalent with owl:equivalentClass or A_1 is a subclass of A_2 (the case when A_2 is a subclass of A_1 is analogous), then $cf_p = \frac{1}{a_1 + 1}$. Otherwise, $df_p^1 = \frac{1}{a_1 + 1}$ and $df_p^2 = \frac{1}{a_2 + 1}$.

2. The property *p* is defined with owl:someValuesFrom for both O_1 and O_2. Let the property *p* be defined for O_1 with

`<owl:someValuesFrom rdf:resource="#`B_1`">`
and for O_2 with
`<owl:someValuesFrom rdf:resource="#`B_2`">.`

Let b_1 (resp. b_2) be the number of subclasses of B_1 (resp. B_2) and *w* be the number of classes in the whole domain. If B_1 and B_2 are equal classes or declared equivalent with owl:equivalentClass or B_1 is a subclass of B_2 (the case when B_2 is a subclass of B_1 is analogous), then $cf_p = \frac{1}{(b_1 + 1)w}$. Otherwise, $df_p^1 = \frac{1}{(b_1 + 1)w}$ and $df_p^2 = \frac{1}{(b_2 + 1)w}$.

3. The property *p* is defined with owl:hasValue for both O_1 and O_2. Let the property *p* be defined for O_1 with

`<owl:hasValue rdf:resource="#`V_1`">`

Copyright © 2015, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

and for O_2 with
`<owl:hasValue rdf:resource="#`V_2`">.`

If V_1 and V_2 are the same values or declared equivalent with owl:sameAs, then $cf_p = 1$. Otherwise, $df_p^1 = 1$ and $df_p^2 = 1$.

4. If the property *p* is defined for O_1 with

`<owl:hasValue rdf:resource="#`V_3`">`
and for O_2 with
`<owl:allValuesFrom rdf:resource="#`A_3`">,`

V_3 A_3 $cf_p = 1$ a_3 A_3 $df_p^1 = 1$ $df_p^2 = \frac{1}{a_3+1}$ 5. If the property *p* is defined for O_1 with

`<owl:hasValue rdf:resource="#`V_4`">`
and for O_2 with
`<owl:someValuesFrom rdf:resource="#`B_3`">,`

V_4 B_3 $cf_p = 1$ b_3 B_3 $df_p^1 = 1$ $df_p^2 = \frac{1}{(b_3+1)w}$

6. If the property *p* is defined for O_1 with

`<owl:allValuesFrom rdf:resource="#`A_4`">`
and for O_2 with
`<owl:someValuesFrom rdf:resource="#`B_4`">,`

a_4 b_4 A_4 B_4 $cf_p = \frac{1}{(a_4+1)(b_4+1)w}$ $df_p^1 = \frac{1}{a_4+1}$ $df_p^2 = \frac{1}{(b_4+1)w}$

Further, we consider three kinds of cardinality restriction declarations:

• owl:minCardinality;
• owl:maxCardinality;
• owl:cardinality.

We can distinguish the following cases:

1. If the property *p* is defined with owl:maxCardinality for both O_1 and O_2, and it has value *m* in O_1 and value *n* in O_2, where $m \le n$, then $cf_p = \frac{1}{m}$, $df_p^1 = 0$ and $df_p^2 = n - m$. The case when $m \ge n$ is analogous.
2. If the property *p* is defined with owl:minCardinality for both O_1 and O_2, the values for this restriction would not contribute to common and distinctive features, since each of these

Copyright © 2015, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

restrictions can have infinitely many values. It only contributes to the similarity calculation if it is declared together with owl:maxCardinality restriction, which is then the following case.

3. If the property *p* is defined with owl:cardinality for both O_1 and O_2, and it has value *m* in O_1 and value *n* in O_2, then if $m = n$, then $cf_p = 1$. Otherwise, if $m < n$, then $df_p^1 = 0$ and $df_p^2 = n - m$. The case when $m > n$ is analogous.

In the above, the subclass relation should be taken into account, so that for each class we account for the property definitions inherited from parent classes.

For classes declared equivalent with owl:equivalentClass we assume their similarity based on properties is equal to 1.

3.2. Similarity among Instances

Next, we compare instances of classes by simply comparing the property-value pairs for each instance. If the property *p* has *h'* different values in O_1 and *h"* different values in O_2, and *k* is the number of times O_1 and O_2 have the same value for *p,* then $cf_p = \frac{k^2}{h'h''}$, $df_p^1 = \frac{h'-k}{h'}$ and $df_p^2 = \frac{h''-k}{h''}$.

3.3. Similarity among Classes Defined as Restrictions and Instances

Finally, we look at the procedure for calculating similarity between a class O_1 and an instance O_2. We distinguish the following cases:

1. The property *p* is defined with owl:allValuesFrom for O_1 and has *h* different values in O_2. Let the property *p* be defined for O_1 with

`<owl:allValuesFrom rdf:resource="#`A_1`">.`

If the values in O_2 are not instances of A_1, then $sim(O_1, O_2) = 0$. Otherwise, if a_1 is the number of subclasses of A_1, then $cf_p = \frac{h}{a_1 + 1}$ and $df_p^1 = \frac{a_1 + 1 - h}{a_1 + 1}$.

2. The property *p* is defined with owl:someValuesFrom for O_1 and has *k* different values in O_2. Let the property *p* be defined for O_1 with

`<owl:someValuesFrom rdf:resource="#`B_1`">.`

If the values in O_2 are not instances of B_1, then $sim(O_1, O_2) = 0$. Otherwise, if b_1 is the number of subclasses of B_1, then $cf_p = \frac{k}{(b_1 + 1)w}$ and $df_p^1 = \frac{b_1 + 1 - k}{(b_1 + 1)w}$.

Copyright © 2015, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

3. The property *p* is defined with owl:hasValue for O_1 and has *d* different values in O_2. Let the property *p* be defined for O_1 with

```
<owl:hasValue rdf:resource="#V1">.
```

If V_1 is not among the *d* values in O_2, then $sim(O_1, O_2) = 0$. Othervise, $cf_p = \frac{1}{d}$ and $df_p^1 = \frac{d-1}{d}$.

Repeating the above process for each property defined for O_1 and O_2 we obtain all common and distinctive features of O_1 and O_2:

$$cf(O_1, O_2) = \sum_{i=1}^{k} cf_{p_i}$$

$$df(O_1) = \sum_{i=1}^{k} df_{p_i}^1$$

$$df(O_2) = \sum_{i=1}^{k} df_{p_i}^2$$

where *k* is the number of properties defined for O_1 and O_2. Finally, we calculate the similarity between two objects O_1 and O_2 using the formula (2):

$$sim_T(O_1, O_2) = \frac{cf(O_1, O_2)}{df(O_1) + df(O_2) + cf(O_1, O_2)}$$

This method for property-based similarity of objects was first introduced in (Cena et al., 2012), but only for value restrictions. It was further developed to include cardinality restrictions and applied to categorization of shapes in (Likavec, 2013).

3.4. Property-based Relatedness of Domain Objects

At the beginning of this section we dealt with *similarity* among domain objects, which can be used to find the objects *similar* to each other, i.e. the objects which have the same values for certain properties. Another concept we want to introduce is that of *relatedness* which makes it possible to find the objects which are in a certain way *related* to each other, but not necessarily similar. Our hypothesis is that, in order to calculate the property-based relatedness of domain objects, we can apply similar reasoning as for calculating similarity, but considering the values that are the same, for *different* properties. For example, if for a certain event the property has_main_actor has value Madonna and for another event the property has_director has value Madonna, these two events might not be similar but are definitely related to each other and might be of interest for users.

Copyright © 2015, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

Hence, the question arises: how to determine the properties which would be used in calculating relatedness of objects? In some domains, it is certainly possible to use all the properties, knowing that if they have common values, this would be enough to relate the corresponding objects which have these properties. In more complex realistic domains, it might make sense to calculate relatedness only on properties related or connected in a certain way, since considering completely unrelated properties would lead to relating very different objects. To do so, it can be necessary to group the properties using some criteria, and then calculate the relatedness using the properties belonging to the same group.

There are various approaches to this grouping of properties. One possibility is to organise the properties into sub-ontologies and then consider properties that have the same ancestor at the certain level of generality. For example, in the events domain, one option would be to have the property has_performer which has as sub properties has_main actor, has_singer, has_goalkeeper etc. Then, the related events would all have the same values for the properties which have has performer as the ancestor. Similarly, the properties which are somehow related can be grouped together by ontology designers. Another directive would be to not consider the inverse properties in the same group. Another possibility is to group all the properties with the same domain.

In what follows, we would assume that $G_1,...,G_s$ are the groups of related properties. These groups can have overlapping properties, i.e. it is possible to find one property in more than one group. Note that we might end up having only one group, i.e. all the properties from the domain, if we decide not to impose any categorization.

When calculating relatedness of two domain objects O_1 and O_2 for a given group of properties G_i, we exclude the pairs where one property has the same values for O_1 and O_2, since this case has already been accounted for to calculate their similarity. Then we apply the same algorithm used for similarity calculation taking into account all the properties from the group G_i. In this way, we find all the common and distinctive features w.r.t. the properties of the group G_i rather than w.r.t. a single property. The process is analogous for all other groups $G_1,...,G_s$ if present.

Let us denote with $rel_{G_i}(O_1,O_2)$ the value of semantic realatedness of objects O_1 and O_2 w.r.t. the group $G_i, i=1,...,s$. Then the semantic relatedness of O_1 and O_2 is calculated as a sum of all the corresponding relatedness measures:

$$rel(O_1,O_2)=\sum_{i=1}^{s} rel_{G_i}(O_1,O_2).$$

3.5. Complexity for the Computation of Similarity/Relatedness Measures

We will show how to determine the complexity of similarity measure computation for two instances O_1 and O_2, the other cases being similar. We will assume that *s* is the maximum number of properties defined for an object and *n* is the maximum number of values for a certain property. When calculating the similarity between two instances, we need to calculate common features for these two instances, as well as distinctive features for both of them.

Let us consider first the property p_i which two instances O_1 and O_2 have in common. We will denote by $O_1.p[i]$ the property p_i in O_1 and by $O_2.p[i]$ the property p_i in O_2. In order

Copyright © 2015, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

Algorithm 1. Calculate CF for property p

```
1: function calculateCF (O1, O2, propId);
2: V′ = getValues(O1, O1.p[propId]);
3: V″ = getValues(O2, O2.p[propId]);
4: k = intersection(V′, V″).size;
5: CF = k2 / V′.size * V″.size;
6: return CF;
```

to calculate common features for p_i we need to find the cardinality of intersection of the sets which contain the values for p_i for both objects, whereas to calculate the distinct features for both we need to find the cardinalities of corresponding set differences. This is illustrated with the following code snippet:where *V'* is the set of all the values for *p* in O_1 and *V"* is the set of all the values for *p* in O_2. We can distinguish three cases:

- The values in *V'* and *V"* are sorted. Then the complexity of calculation of $cf_p(O_1,O_2)$ (i.e. of the intersection of the two sets) is $\Theta(V'.size+V''.size)$. If n is the maximum number of values for a certain property we can conclude that this complexity is $O(n)$. The calculation of $df_p^1(O_1,O_2)$ and $df_p^2(O_1,O_2)$ has the same complexity.
- The values in *V'* and *V"* are not sorted and we sort them first. Then the complexity is $O(n\log(n))$ for $cf_p(O_1,O_2)$, $df_p^1(O_1,O_2)$ and $df_p^2(O_1,O_2)$.
- The values in *V'* and *V"* are not sorted and we do the above calculation directly. Then the complexity is $O(n^2)$ for $cf_p(O_1,O_2)$, $df_p^1(O_1,O_2)$ and $df_p^2(O_1,O_2)$.

Next, we have to repeat this process for each property *p* that objects O_1 and O_2 have in common. The complexity of finding all the common properties for these two objects is $\Theta(s)$, where *s* is the maximum number of properties defined for an object.

Hence, computing the similarity between these two objects calculates $cf_p(O_1,O_2)$, $df_p^1(O_1,O_2)$ and $df_p^2(O_1,O_2)$ for all the properties these objects have in common as can be seen below (where we assume that the values in *V'* and *V"* are sorted, the other two cases being analogous):where *P'* is the set of all the properties defined for O_1, *P*" is the set of all the properties defined for O_2 and *common* is the set of properties O_1 and O_2 have in common. Hence the final complexity when *V'* and *V"* are sorted is $O(s+sn)=O(sn)$. In case *V'* and *V"* are not sorted but we sort them, it is $O(sn\log n)$ and in case they are not sorted and we calculate the common and distinctive features directly, it is $O(sn^2)$.

Note that normally the values for s and n would not reach high values. It is hard to immagine that even one of them would go over 100 since they are the number of properties defined in the ontology and the maximum number of values for a certain property.

When calculating relatedness, we group the related properties and then repeat the same process as for calculating similarity, with these groups of properties.

Copyright © 2015, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

Algorithm 2. Calculate similarity of instances

```
function calculateSimilarity(O1, O2)
P′ = getProperties(O1);
P″ = getProperties(O2);
common = intersection(P′, P″); # complexity $\Theta(s)$
den = 0;
num=0;
for all propId in common do
    CF[propId] = calculateCF(O1, O2, propId); # complexity $O(n)$
    DF1[propId] = calculateDF1(O1, O2, propId); # complexity $O(n)$
    DF2[propId] = calculateDF2(O1, O2, propId); # complexity $O(n)$
    num += CF[propId];
    den += CF[propId] + DF1[propId] + DF2[propId]; # complexity $O(3n) = O(n)$
end for
sim = num / den; # complexity $O(sn)$
return sim;
```

4. RELEVANCE OF PROPERTIES

Not all the features have the same importance in defining a concept in a domain, and this different relevance should be taken into account in the similarity computation.

The relevance of properties can be declared *a priori* by domain experts, as we did in our previous work (Cena et al., 2012). This solution is effective, but it may not be very feasible for a huge domain. Thus, in this work we wanted an automatic method to determine the relevance of properties. We started from the assumption that the relevance of a property can be considered as the capacity of the property to determine the similarity between two entities. For example, it can be argued that for the concept Book the property has_editor is probably less relevant than the property has_author, since two books of the same author would be considered more similar than two books of the same editor. Therefore, the relevance factor for has_author should be higher then the one for has_editor.

This intuition suggests the following procedure to automatically learn the relevance of properties in our ontology (see Algorithm 3).

The algorithm takes as input a collection of items and the domain ontology. It returns a relevance factor in the range $[0-1]$ for each property, representing its relevance. First, it computes the items similarity by using our similarity measure described in Section 3. Then, it calculates the relevance factor for each property as the square of the average similarity between items with the same value for that property. For example:

```
sim(Evita_musical, Evita_movie) = 0.7
sim(Desperately_Seeking_Susan, Evita_movie) = 0.6
sim(Matador, Evita_movie) = 0.5.
```

All these domain objects have the property has_main_actor with the same value (either pairs with value Madonna or pairs with value Antonio_Banderas.) Hence, the average similarity among the domain objects which have the property has_actor defined for them is 0.7+0.6+0.5 = 0.6. So the relevance factor for the property has_actor is $R_{has\_actor} = 0.6^2 = 0.36$.

Copyright © 2015, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

Algorithm 3. Learn property relevance

```
1: function learnPropertyRelevance(items, ontology)
2:   for all i,j in items do
3:     properties = getProperties(i, j, ontology);
4:     for all p in properties do
5:       if haveSameValue(p, i, j) then
6:         counter[p]++; # increment property p counter
7:         sumSim[p]= sumSim[p] + similarity[i][j]); # add the similarity to sumSim
8:       end if
9:     end for
10:  end for
11:  for all p in properties do
12:    relevances[p]= (sumSim[p] / counter[p])^2 ; # compute the relevance
13:  end for
14:  return relevances;
```

The square value of the average similarity proved useful in the preliminary tests by assigning a nonlinear bonus to the highest results. Since it is a statistical technique, we should take into consideration only the properties for which a good number of pairs share the same value. When this is not true, any inference about the relevance is too risky and the property should be assigned a neutral relevance factor. For this reason in the prototype we assign to the properties with counter[p] < n (with *n* set empirically to 5) a relevance factor equal to 0.25, which we consider as neutral value, since it is equivalent to the relevance value of a property which has the same value for items that have an average similarity of 0.5 out of 1.

For example, in the events dataset we used for the evaluation (Section 6), the pairs of events which share the same value for the property has_food_type have the average similarity of 0.84, and thus, this property will have a relevance factor of $R_{has\_food\_type} = 0.84^2 = 0.7$. Conversely, the pairs which share the same value for the property has_price have the average similarity of 0.7, resulting in relevance factor of $R_{has\_price} = 0.49$. Intuitively, this means that, according to a certain sample of users, two events are more similar if they offer the same kind of food than if they are in the same price range.

We then consider a *relevance factor* R^p for each property *p* in our Formula 2 for calculating the mutual similarity between domain objects O_1 and O_2 in order to weight it accordingly during the propagation process. Thus, the Formula 2 becomes:

$$sim_T^r(O_1,O_2) = \frac{cf^r(O_1,O_2)}{df^r(O_1) + df^r(O_2) + cf^r(O_1,O_2)}$$

where

Copyright © 2015, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

$$cf^r(O_1,O_2)=\sum_{i=1}^{k}R_i cf_{p_i}$$

$$df^r(O_1)=\sum_{i=1}^{k}R_i df_{p_i}^1$$

$$df^r(O_2)=\sum_{i=1}^{k}R_i df_{p_i}^2\,.$$

Another approach to calculate the relevance of properties would be to use different techniques for similarity calculation such as item to item collaborative filtering approach (CF) (Sarwar et al., 2001). In CF the similarity between two items is calculated as cosine similarity between user ratings for these two items, whereas in our case it depends on the properties defined for the objects in the ontology. We chose not to use a collaborative filtering approach since in a cold start phase similarity derived from user ratings would not be feasible: the sparsity of ratings would make it impossible to compute the similarity for the majority of pairs of items. As the number of ratings grows and therefore the similarity between items becomes more accurate, the similarity derived from user ratings could be used in combination with our approach to improve the calculation of relevance factors.

5. PROPAGATION OF USER INTERESTS

For propagating user interests in the user models we adopt a slight modification of the approach presented in (Cena et al., 2012). The propagation of user interests in the domain ontology allows to infer the missing information about the user preferences, starting from the initial set of interest values.

We first give details of how we model users, followed by the key steps of our approach and the details of the propagation process.

5.1. User Model

One requirement of our approach is the representation of the domain by means of OWL ontologies, with explicit specification of the concept properties. The user model contains the values of user interest for the objects in the ontology. Thus, the user model is defined as an overlay over such an ontology (the so called *ontology-based user model* (Brusilovsky, 1996)). In this case, the ontological user profile is an instance of the domain ontology, with an interest value associated to each concept or instance. Usually, each node *N* of the domain ontology is a pair <*N, I(N)*>, where *I(N)* is the interest value for the object *N*. In our case, for each object *N* we have instead a quadruplet $<N,I_S(N),I_R^{sim}(N),I_R^{rel}(N)>$ where:

- $I_S(N)$ is the sensed interest, derived from the direct feedback for the object *N*,
- $I_R^{sim}(N)$ is the *total interest received from similar objects*,
- $I_R^{rel}(N)$ is the *total interest received from related objects*.

Copyright © 2015, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

This solution allows for a great flexibility, since the system can implement different policies to combine the three interest values together according to context and user needs. For example, it can choose to use a combination of sensed interest and propagated interest from similar objects when it is needed to maximize the accuracy of the suggestions, whereas it may prefer to include also the propagated interest received from related objects in order to yield higher diversity. Section 6 will give more details on the peculiarity of the different combinations of interests. In most cases, a solution which combines all these three factors seems to be the most effective.

At the beginning of the interaction the user model is empty and the interest values are inserted as soon as the system obtains some feedback from users. The users can provide their interests in domain objects by rating them or this interest could be automatically inferred by the systems based on users behavior, recording the user actions such as browsing, clicking, etc. However, how the user interests are gathered by the system is out of the scope of this paper.

5.2. Propagation Process

We summarize here the key steps of our approach which we then explain in more detail below:

1. Precompute the *similarity* and *relatedness* among all the objects in the ontology (see Sections 3 and 4).
2. Register the *user feedback* for the objects in the ontology (Section 5).
3. For each object *N* receiving feedback from user *U*:
    - Calculate the *sensed interest value* for *N* and store it in the user model of *U*. Sensed interest depends on the direct user feedback for the object and the level of the object in the conceptual hierarchy derived from the ontology.
    - Calculate the *propagated interest value* propagated from *N* to all the objects similar and related to it, and update their *total interest received from similar objects* and *total interest received from related objects* in the user model of *U*. Propagated interest values depend on the original sensed interest of *N* and on the similarity/relatedness between *N* and the object receiving the propagated value.
    - For each subsequent user feedback, propagate the difference between the newly obtained sensed interest value for *N* and the old one.

The total interest value for object *N* for a certain user is computed by the combination of three values: the *sensed interest value* $I_S(N)$, the *total interest received from similar objects* $I_R^{sim}(N)$ and the *total interest received from related objects* $I_R^{rel}(N)$. We explain below how to compute all the above interest values.

5.2.1. Sensed Interest Value

When a user provides a feedback for a certain object in the domain, we first calculate the *sensed interest* for that object, which captures the object's reaction or sensitivity to the user feedback. Sensed interest for the object *N* depends on its position in the conceptual hierarchy derived from the domain ontology and the feedback provided by the user and is calculated as follows:

$$I_S(N) = \frac{\ell(N)}{\max} sig(F(N) + f(N)) \tag{3}$$

Copyright © 2015, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

where

- $\ell(N)$ is the level of the object receiving the feedback,
- max is the level of the deepest object in the conceptual hierarchy,
- *sig*(*x*) is a variant of the sigmoid function given by $sig(x) = \frac{2}{1+e^{-x}} - 1 = \frac{e^x - 1}{e^x + 1}$,
- *F*(*N*) is the sum of the previous feedback values obtained from the user for the given object *N*,
- *f*(*N*) is the current (new) feedback obtained from the user for the object *N*.

The *sig*(*x*) function was introduced to modulate the subsequent feedback values received by an object as a result of multiple actions on the same object (e.g. clicking, tagging, sharing). Notice that the position of the concept in the conceptual hierarchy reflects the different levels of generalization present in the hierarchy (Resnik, 1999): the concepts lower down in the conceptual hierarchy represent more specific notions, and as such signal more precise interests than the concepts represented by upper classes in the conceptual hierarchy, which express more general notions.

5.2.2 Propagated Interest Value

The sensed interest value is further used for the subsequent propagation phase, in which the interest is propagated to similar and related objects. We use the properties of each domain object to calculate its similarity (Section 3) and relatedness (Section 3) with other domain objects. Since similarity and relatedness between objects depend only on the given ontology and the properties defined for its objects, they are precomputed and subsequently updated only when the ontology is modified. This propagation does not have any particular direction (as opposed to the one described in Cena et al., 2013 and allows the propagation of the user interests to various (sometimes quite distant) objects in the ontology.

We distinguish the initial user feedback and the subsequent ones. For the initial user feedback, we compute the *initial propagated interest value* $I_P^{sim,0}(N,M)$ from the node *N* which received the feedback to the *similar* node *M*, using the hyperbolic tangent function as follows:

$$I_P^{sim,0}(N,M) = \frac{e^{2sim(N,M)} - 1}{e^{2sim(N,M)} + 1} I_S(N) \tag{4}$$

and similarly the *initial propagated interest value* $I_P^{rel,0}(N,P)$ from the object *N* which received the feedback to the *related* object *P*:

$$I_P^{rel,0}(N,P) = \frac{e^{2rel(N,P)} - 1}{e^{2rel(N,P)} + 1} I_S(N) \tag{5}$$

where

- $I_S(N)$ is the sensed interest of the object *N* receiving the feedback;

Copyright © 2015, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

- $sim(N,M)$ is the *similarity* between the object *N* receiving the feedback and the object *M* receiving the propagated interest (see Section 3);
- $rel(N,M)$ is the *relatedness* between the object *N* receiving the feedback and the object *P* receiving the propagated interest (see Section 3).

Only with the initial user feedback for a certain domain object, all the sensed interest is used in the propagation process. With each subsequent *i*-th user feedback, the sensed interest used to update the user model is obtained as a difference between the newly obtained sensed interest value for the given object and the old one that was already propagated:

$$\Delta I_S^i(N) = I_S^i(N) - I_S^{i-1}(N) = \frac{\ell(N)}{\max}\left(sig(F(N)+f(N)) - sig(F(N))\right) \tag{6}$$

Therefore, for each subsequent *i*-th feedback for the object *N* (after the initial one), the propagated interest $\Delta I_P^{sim,i}(N,M)$ to the similar node *M* and the propagated interest $\Delta I_P^{rel,i}(N,P)$ to the related node *P* is calculated as:

$$\Delta I_P^{sim,i}(N,M) = \frac{e^{2sim(N,M)}-1}{e^{2sim(N,M)}+1}\Delta I_S^i(N) \tag{7}$$

$$\Delta I_P^{rel,i}(N,P) = \frac{e^{2rel(N,P)}-1}{e^{2rel(N,P)}+1}\Delta I_S^i(N) \tag{8}$$

The amount of interest propagated for the *i*-th feedback is lower than the amount of interest propagated for any previous feedback. This balances the system since no repeated or redundant information is propagated.

Since for the initial feedback, there is no previous feedback, we obtain $\Delta I_S^0(N) = I_S(N)$, $I_P^{sim,0}(N,M) = \Delta I_P^{sim,0}(N,M)$ and $I_P^{rel,0}(N,P) = \Delta I_P^{rel,0}(N,P)$.

Hence, we can always use the values $\Delta I_P^{sim,j}(N,M)$ and $\Delta I_P^{rel,j}(N,P)$ when updating the user model with propagated interest.

So the total propagated interest value from the object *N* to a similar object *M* is the sum of all the propagated interest values from *N* to *M*:

$$I_P^{sim}(N,M) = \sum_{i=1}^{d_N} \Delta I_P^{sim,i}(N,M) \tag{9}$$

where d_N is the number of propagations from *N* to *M*. Similarly, the total propagated interest value from the object *N* to a related object *P* is the sum of all the propagated interest values from *N* to *P*:

Copyright © 2015, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

$$I_P^{rel}(N,P) = \sum_{i=1}^{b_N} \Delta I_P^{rel,i}(N,P) \tag{10}$$

where b_N is the number of propagations from *N* to *M*. Note that d_N and b_N depend on *N*, i.e. each node *N* has a different number of propagations based on the user feedback.

5.2.3. Total Received Interest Value

Finally, we can calculate the *total interest received from similar objects* for the object *O* as the sum of all the propagated interest values from similar nodes:

$$I_R^{sim}(O) = \sum_{j=1}^{n} I_P^{sim,j}(N_j,O). \tag{11}$$

Analogously, we can calculate the *total interest received from related objects* for the object *O* as the sum of all the propagated interest values from related nodes:

$$I_R^{rel}(O) = \sum_{j=1}^{m} I_P^{rel,j}(M_j,O) \tag{12}$$

where $N_1,\ldots,N_n$ are the objects similar to *O* and $M_1,\ldots,M_m$ are the objects related to *O*.

Total interest value For each object *O* we have three different interest values:

- Its sensed interest $I_S(O)$;
- The total interest received from similar objects $I_R^{sim}(O)$;
- The total interest received from related objects $I_R^{rel}(O)$.

These three values can be combined to obtain the total interest value for each object, which can include only the interest values received from similar objects or only from related objects or both.

It should be noted that, since the system can trivially recompute the amount of feedback propagated from object *N* to every other object with formulas (9) and (10), this approach supports *dynamic ontology maintenance* (Beydoun et al., 2011). In other words, it is possible to modify the ontology, for example enriching it by adding a certain property, and then update the interest values for all the domain objects as if the system had always used the modified ontology.

6. EVALUATION

We evaluated our approach using two datasets. The first evaluation is in the domain of event recommendation, whereas the second one concerns the domain of recipes.

Copyright © 2015, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

6.1. Events

The first evaluation uses the event ontology for which the basic event hierarchy is depicted in *Figure 1*. For better clarity *Figure 1* presents only the taxonomy with the classes hierarchy, mitting all the properties of classes. The whole ontology consists of 77 classes (among which 17 classes represent events, whereas other classes represent concepts such as SportType or VenueType), 129 instances (out of which 44 instances of events) and 23 properties (15 object properties and 8 data properties). We considered three kinds of events in the event domain: cultural events (cinema, concert, exhibition, theater, book presentation), gastronomical events (cooking course, lunch, dinner, degustation, aperitivo, food fair) and sport events (race, match, show, open day). This event ontology is not trying to categorise events into a deep event hierarchy, rather it describes each event with its properties. Hence, the similarity and relatedness of events based on properties does not depend only on the position of the item in the ontology graph but exclusively on properties which were used to define subclasses as restrictions. ("Aperitivo" is a traditional way to get together among Italians. In the late afternoon they go for a drink and get served some light food with it. After that they are ready to go for dinner, but sometimes a good aperitivo can substitute a dinner.)

Some of the properties which were defined for the events are: has_venue (specific places where events take place, such as a gallery or a music hall), has_price, has_physical_location (sites of the city of Torino and its surroundings where events take place), has_main_theme, has_main_protagonist, has_tags, etc. Moreover, some events have additional properties, such as type_of_tournament for sport events or has_director for movies. We believe that this ontology encompasses a great variety of events, covering many different areas. Since it contains both, the hierarchy of classes (some of which defined as property restrictions), as well as explicitly defined properties of domain objects, it was a good starting point for calculating semantic similarity and relatedness of domain objects and subsequently evaluating both accuracy and diversity of recommendation. Moreover, since this ontology includes three different sub-domains (cultural events, gastronomic events and sport events), it also allows us to test the performance of our approach also in a cross domain scenario (Fernández-Tobìas et al., 2012).

6.1.1. Hypotheses

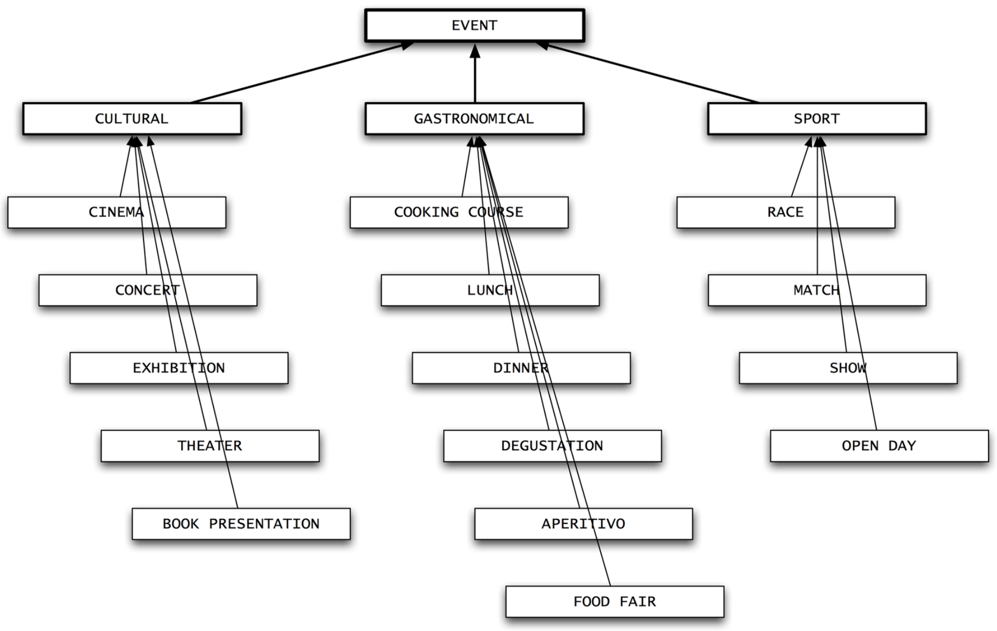
We wanted to verify that:

H1: property-based user interest propagation using similarity provides better recommendation accuracy than the classic edge-based propagation presented in our previous work (Cena et al., 2013) and in other state of the art approaches (Middleton et al., 2004, Sieg et al., 2007);
H2: assigning a certain weight or relevance (as described in Section 4) to each property brings a significative improvement to the accuracy of recommendations;
H3: property-based similarity offers good results w.r.t. diversity. In addition, considering relatedness in user interest propagation (as described in Section 3) yields a good increment to recommendations diversity, paying a relatively low cost in accuracy;
H4: property-based user interest propagation is particularly effective in improving accuracy when propagating user interest values across different sub-domains to provide cross-domain recommendation.

In this evaluation we compare accuracy and diversity of five approaches for user interest propagation, namely:

Copyright © 2015, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

Figure 1. Event taxonomy



1. Edge-based user interest propagation (as in (Cena et al., 2013)), labelled E. We used the conceptual distance described in (Cena et al., 2013) and propagated to all items with the distance ≤ 1. We selected this threshold since it maximizes the accuracy according to the preliminary tests.
2. Property-based user interest propagation using *similarity* only, labelled P;
3. Property-based user interest propagation using both *similarity and relatedness*, labelled PR;
4. Property-based user interest propagation using *similarity* and assigning *relevance* to properties, labelled PW;
5. Property-based user interest propagation using *similarity and relatedness* and assigning *relevance* to properties, labelled PWR.

6.1.2. Evaluation Setting and Measures

In order to collect users' preferences, we designed a web questionnaire in which users anonymously filled in their preferences (on a scale from 1 to 10) for 44 heterogeneous events, including concerts, art exhibitions, soccer games, food festivals and many others. The ratings collected from the users were recorded in a database.

6.1.3. Measures

We measure the accuracy in predicting the correct user preferences by using two metrics: Spearman's rank correlation coefficient ρ and Mean Absolute Error (MAE). Spearman's rank correlation ρ provides a non-parametric measure of statistical dependence between two ordinal variables and gives an estimate of their relationship using a monotonic function. In the absence of repeated data values, a perfect Spearman correlation of $\rho = +1$ or $\rho = -1$ is obtained when each of the variables

Copyright © 2015, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

is a perfectly monotonic function of the other, either direct or inverse. Tied values are assigned a rank equal to the average of their position in the descending order of values. Conventionally, $\rho \geq 0.5$ points to a "fair direct correlation". The Mean Absolute Error (MAE) is a classic metric used to express the distance between predicted and effective outcome. In our case it is computed as the average difference between the user ratings of an item and the estimated one.

The performance of the five algorithms was compared by means of the chi-squared test to make evident any statistically significant difference. As common practice, we consider two approaches significantly different if the chi-squared test yields a null hypothesis $p > 0.05$. The most interesting findings are summarized in Table 1.

We measure the recommendation diversity by counting the different numbers of categories recommended to users.

6.1.4. Subjects

The initial sample included 247 subjects, 19-33 years old, recruited according to an availability sampling strategy[7] among the students of various courses (such as Business Intelligence, HTML, Web of Things Design) at the University of Torino, Italy, as well as among authors' acquaintances. Even though the best way for having a representative sample is random sampling, these strategies are time consuming and not always financially justifiable. Therefore, much research in social science is based on samples obtained through non-random selection, such as the availability sampling, i.e. a sampling of convenience, based on subjects available to the researcher, often used when the population source is not completely defined.

We restricted the initial sample to 231 subjects (10164 ratings) by eliminating all the subjects who assigned the same vote to more than 80% of the items, since this was a strong indicator of a random preference assignment. The dataset used is available from the authors upon request for academic purposes.

To verify the hypotheses H1 and H2, we assessed the performance of the five approaches, splitting the feedback set into training set and test set. The training set was used as input for user interest propagation, whereas the test set was compared with the suggested items. We performed different tests, changing the percentage of feedback values in the training set from 10% to 70%. We ran each test 10 times and in each run we selected randomly which feedback values to be inserted in the training set. Finally, we computed the average Spearman correlation coefficient ρ and MAE.

6.1.5. Accuracy Test

Figure 2 shows the average Spearman's rank correlation ρ for increasing percentage of feedback values in the training set. In all the cases PW obtains the highest accuracy, followed by PWR and then by P, PR and E. In particular, both P and PW performed significantly better than the baseline edge-based user interest propagation in each test, according to the chi-squared test (see Table 1 for the values of ρ for the different tests). Actually, the difference is so big that PW accuracy in the 10% case is still higher than E accuracy when taking as training set 70% of the feedback values.

As expected, PWR yields a slightly lower accuracy than PW, but the difference between these two approaches is never statistically significant. In a similar way, PR performs slightly worse than P, but the difference between these approaches is significant only for the 70% case. Hence, we can conclude that, even if the relatedness appears to reduce the accuracy, its effect is in most cases negligible, while at the same time contributing to better diversity (see Diversity test).

Table 1 shows the percentage of users with $\rho \geq 0.5$ and $\rho < 0$ obtained by the five algorithms in the different tests. PW provides the best results in all tests, yielding both the highest number

Copyright © 2015, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

Table 1. Summary of the main results of the evaluation (in parenthesis the 95% confidence interval). In bold the best result of each test (less is better for ρ<0, more is better for ρ ≥ 0.5)

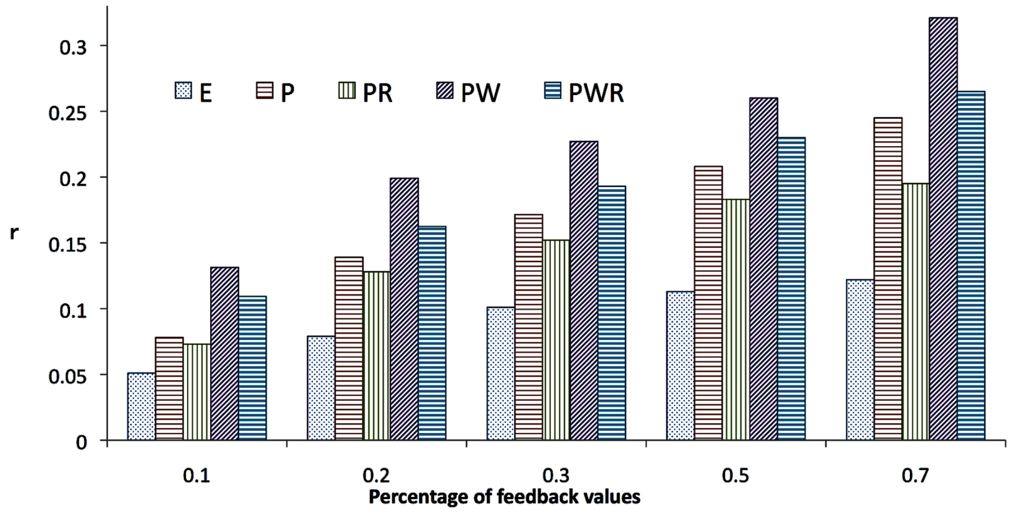
Percentage of users with ρ ≥ 0.5					
Test	**E**	**P**	**PR**	**PW**	**PWR**
10%	3.0% (1.2-6.1)	6.5% (3.7-10.5)	6.1% (3.4-10.0)	**6.9%** (4.0-11.0)	**6.9%** (4.0-11.0)
20%	6.1% (3.4-10.0)	10.4% (6.8-15.1)	9.1% (5.7-13.6)	**13.8%** (9.7-19.0)	9.5% (6.1-14.1)
30%	10.8% (7.1-15.6)	13.0% (8.9-18.0)	11.3% (7.5-16.1)	**18.2%** (13.4-23.8)	14.7% (10.4-20.0)
50%	12.1% (8.2-17.0)	16.0% (11.5-21.4)	13.9% (9.7-19.0)	**23.8%** (18.5-29.8)	18.2% (13.4-23.8)
70%	14.3% (10.0-19.5)	30.3% (24.5-36.7)	23.4% (18.1-29.4)	**39.0%** (32.6-45.6)	33.8% (27.7-40.3)
Percentage of users with ρ < 0					
Test	**E**	**P**	**PR**	**PW**	**PWR**
10%	33.8% (27.7-40.3)	32.9% (26.9-39.4)	33.3% (27.3-39.8)	**29.9%** (24.0-36.2)	32.5% (26.5-38.9)
20%	29.9% (24.0-36.2)	26.4% (20.8-32.6)	26.8% (21.2-33.1)	**19.9%** (15.0-25.7)	23.4% (18.1-29.4)
30%	31.2% (25.3-37.6)	23.4% (18.1-29.4)	25.1% (19.7-31.2)	**19.0%** (14.2-24.7)	21.6% (16.5-27.5)
50%	28.1% (22.4-34.4)	21.2% (16.1-27.1)	21.2% (16.1-27.1)	**19.0%** (14.2-24.7)	22.9% (17.7-28.9)
70%	29.0% (23.2-35.3)	21.6% (16.5-27.5)	23.8% (18.5-29.8)	**16.5%** (11.9-21.9)	19.0% (14.2-24.7)
Chi-square test (in bold the not significant result)					
	10%	**20%**	**30%**	**50%**	**70%**
E vs P	$2.1 \cdot 10^{-3}$	$2.9 \cdot 10^{-5}$	$6.4 \cdot 10^{-5}$	$5.8 \cdot 10^{-7}$	$1.4 \cdot 10^{-7}$
E vs PW	0.0123	$5.5 \cdot 10^{-9}$	$2.2 \cdot 10^{-12}$	$2.3 \cdot 10^{-8}$	$2.2 \cdot 10^{-26}$
E vs PWR	**0.129**	$3.4 \cdot 10^{-6}$	$3 \cdot 10^{-7}$	$3.5 \cdot 10^{-9}$	$1.8 \cdot 10^{-12}$
P vs PR	**0.902**	**0.9**	**0.748**	**0.75**	0.04
P vs PW	**0.134**	$1.8 \cdot 10^{-3}$	0.0329	0.0446	$2.1 \cdot 10^{-5}$
PW vs PWR	**0.359**	**0.079**	**0.37**	**0.08**	**0.255**

of users with ρ ≥ 0.5 and the minimal number of users with ρ < 0. Not only PW is the most effective tool for suggesting lists with a good correlation with users' preferences but it is also the least prone to yielding misleading lists.

The histogram in *Figure 3* shows the MAE values for the five algorithms for different percentage of feedback values in the training set. To compute MAE, we normalised the total interest values to a [0-10] interval, so that they could be compared with user ratings that vary between 0 and 10. When using MAE as evaluation metric, the best algorithm turns out to be PWR: thus it appears that the relatedness allows for a better estimate of user ratings, even if it is less effective in predicting their order in the list of preferences. As before, the disparity between the algorithms grows with the number of feedback values in the training set: the difference between E and PWR is only 0.37 when taking as input 10% of the ratings, but grows to 1.83 when using 70% of them.

Copyright © 2015, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

Figure 2. Average ρ for different percentages of feedback values in the training set

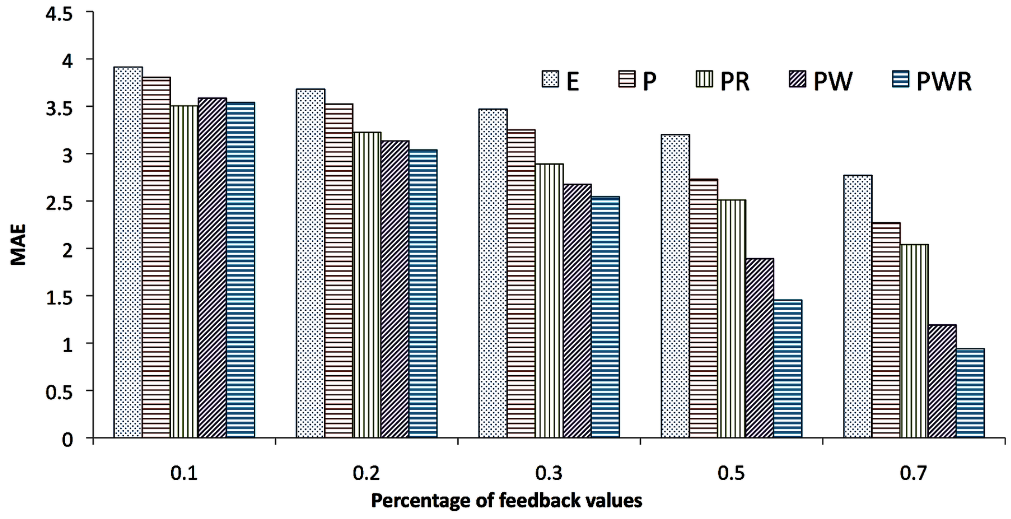


6.1.6. Diversity Test

It is well known that testing accuracy is by itself not enough. Indeed, suggestions that are only slight variations of the same item may be useless for the user. Hence, it is important to check also if the results offer some diversity.

To verify the hypothesis H3, we tested the diversity yielded by the five approaches by generating, for each item in our domain, the list of the most similar *n* items (testing different values for *n*) and then assigning them to fifteen categories, corresponding to the third level classes of the domain taxonomy showed in *Figure 1*. We estimate the diversity of each algorithm as the

Figure 3. Average MAE for different percentages of feedback values in the training set



Copyright © 2015, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

average number of categories represented by the first *n* similar instances. Naturally, we considered only items that could be reached by means of the propagation process, thus in some cases the list of the most similar items may actually include less then *n* items (e.g., when $n = 20$ and a certain item has only 15 similar items).

Figure 4 shows, for each algorithm, the average number of categories (vertical axis) represented in the set of the first *n* most similar objects for each domain object (hori zontal axis). A higher number of categories, in particular for the first 5-10 similar items, points to a greater diversity in the recommendations.

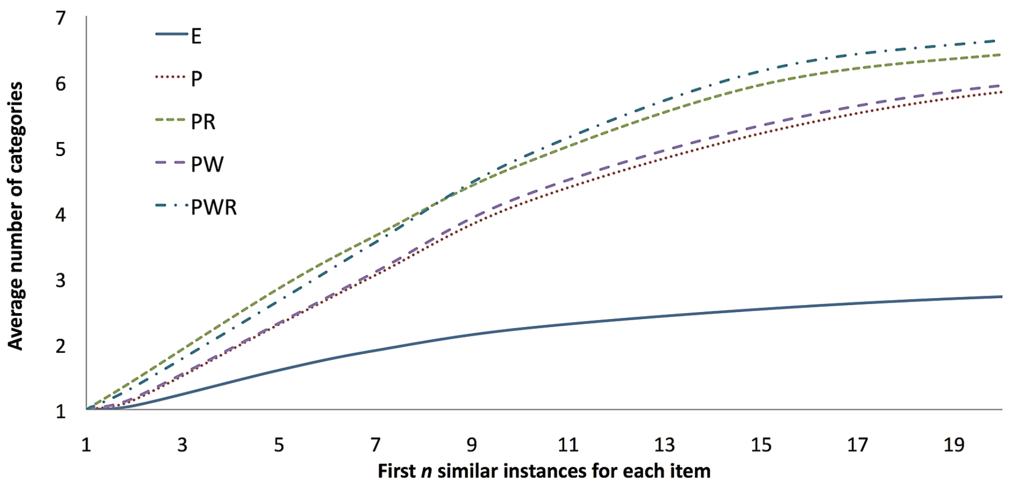
The PR algorithm obtains the highest diversity for $n < 7$, followed by PWR, PW, P and E. For $n \geq 7$ the best algorithm is PWR, followed by PR, PW, P and E. For example, when suggesting 5 items PR yields on average 0.19 categories more than PWR, 0.53 more than PW, 0.55 more than P and 1.25 more than E. When suggesting 15 items, PWR yields on average 0.21 categories more than PR, 0.83 more than PW, 0.95 more then P and 3.63 more than E.

The Wilcoxon test shows that property-based approaches are significantly different from edge-based approach ($p < 0.01$). Better performance of PWR and PR is explained by the presence of property-based relatedness, which was explicitly designed to increment diversity by considering similar values of different properties, thus highlighting implicit relationships between different classes. In fact, PWR and PR are significantly different from PW and P (p<0.02). However, also P and PW have significantly higher diversity than E. In fact, even the simplest property-based user interest propagation (P) has an advantage of 0.8 categories over the baseline edge-based user interest propagation when considering a list of 5 items, lead which grows up to 3.51 categories for a list of 20 elements. It thus appears that an additional advantage of using property-based user interest propagation provides a significant increment in the diversity of the results, which in turn can still be further enhanced by using property-based relatedness.

6.1.7. Cross Domain Test

The recent years experienced the emergence of a variety of cross-domain recommender systems (Fernández-Tobìas et al., 2012), whose aim is to exploit user preferences about certain domains

Figure 4. Diversity: average number of categories represenetd by the first n similar instances for each item



Copyright © 2015, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

in order to suggest items in different domains, often reusing ratings collected by a variety of applications (for instance combining the preferences from the musical and the movie domain). In this section we will show that the techniques illustrated in this paper can be significantly effective also in cross-domain recommendation.

The events ontology spans over three main subdomains: 48% of the events belong to the cultural domain, 32% to the gastronomy-related domain and 20% to the sport domain. To verify the hypothesis H4 and thus to test our approach in a cross-domain scenario, we selected as input the items in one subdomain and tried to infer user interests in the other two subdomains. This choice of subdomains is motivated by the ontology design itself, since these three subdomains represent different contexts for event recommendation. This definition of domain diversity is based on a specific domain and its subdomains, but it can be extended to different domains where the domain objects in all the domains have the same properties with the appropriate values defined. For example, we could combine the movies domain and the books domain and try to infer users' interests in one domain based on their feedback in the other one, since it could be expected that these two domains would have equal or related properties. As *Figure 5* shows, the difference in accuracy between the edge-based user interest propagation (E) and various property-based interest propagation techniques (P, PW, PR and PWR) is even bigger when propagating user preferences across different domains.

6.1.8. Discussion

The evaluation shows that the property-based approaches, while they need a richer ontology to start with, perform much better than the baseline edge-based approaches, as far as accuracy and diversity are concerned (the chi-squared test yielded $0.1 \leq p \leq 2 \cdot 10^{-26}$). Assigning relevance to properties leads also to a significant improvement in accuracy (for 20% of the feedback values or more, $0.03 \leq p \leq 2 \cdot 10^{-5}$).

The algorithm which includes the property-based relatedness yields a significant increment to the diversity of the results ($p < 0.048$ for PR and $p < 0.035$ for PWR), paying a relatively low price in accuracy. Moreover, the cross domain test highlighted that property-based approaches perform much better in a cross-domain scenario than edge-based ones ($p \leq 2 \cdot 10^{-5}$).

PW appears to be the best solution for maximizing accuracy, whereas PR and PRW are respectively the best choices for maximizing diversity for short list ($n < 7$) and medium or long list of suggested items. Since the difference in accuracy between PWR and PW is not statistically significant ($p > 0.8$), we believe that PWR may be the best choice in the majority of realistic situations.

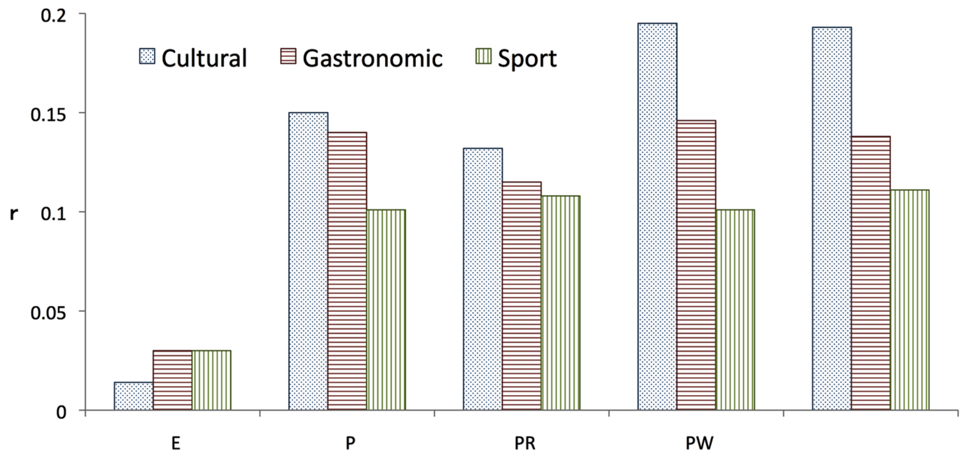
6.2. Recipes

The second evaluation uses the slightly modified Wikitaaable dataset (wikitaaable) used in Cordier et al. (2014). This dataset contains 1666 recipes with properties such as can be eaten as or has_ingredient or suitable_for_diet, providing a good setting to test certain aspects of our approach. We only slightly modified it in the following: in the original ontology all the recipes were instances of Recipe, whereas we wanted to use the hierarchical structure of the ontology (also to be able to test the edge-based approach), hence we used the recipes as instances of various Dish_Type's. In the original ontology dish_type is a property. In *Figure 6*, we show the basic taxonomy of recipes from Wikitaaable, but only including the top categories and the ones from which we used the instances to test our approach. The properties are again omitted for clarity.

The main dish types are: BakedGoodDish, BeverageDish, CrepeDish, MousseDish, PancakeDish, PreserveDish, RollDish, SaladDish, SaltedDish, SauceDish, SoupAndStuffDish,

Copyright © 2015, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

Figure 5. Average ρ when propagating preferences from a single domain



SpecialDietDish, SweetDish, WaffleDish. The recipes we used for our testing did not belong to all recipe groups, since the complexity of the test would be too high and we would rise random answers from the users.

6.2.1. Hypotheses

In this evaluation we could only verify the first three hypothesis, namely:

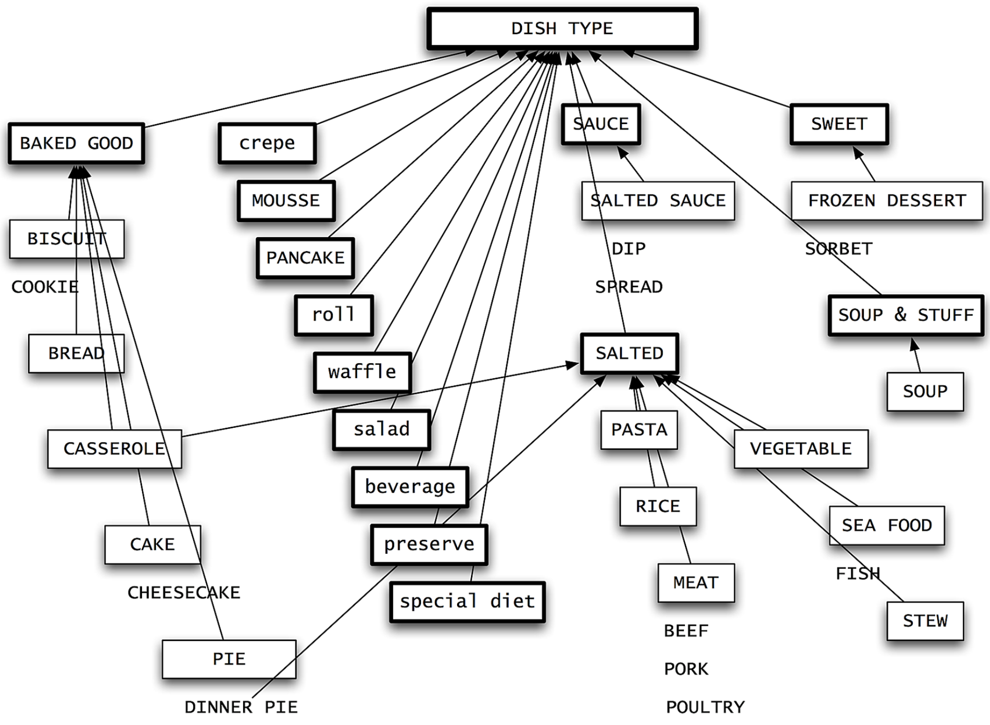
H1: property-based user interest propagation using similarity provides better recommendation accuracy than the classic edge-based propagation;
H2: assigning a certain weight or relevance (see Section 4) to each property brings an improvement to the accuracy of recommendations;
H3: even the simple property-based similarity yields a good increment to recommendations diversity.

In this test we could not test the propagation which takes relatedness into account, due to the definition of properties in the ontology and their relative values. Also, we could not test the cross-domain recommendation since the data set contained only the recipes. Hence, in this evaluation we compare accuracy of three approaches for user interest propagation, namely:

1. Edge-based user interest propagation (as in Cena et al., 2013) using again the conceptual distance, labelled E;
2. Property-based user interest propagation using *similarity* only, labelled P;
3. Property-based user interest propagation using *similarity* and assigning *relevance* to properties, labelled PW.

Copyright © 2015, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

Figure 6. Recipe taxonomy



6.2.2. Evaluation Setting and Measures

In order to collect users preferences, we designed another web questionnaire in which users anonymously filled in their preferences (on a scale from 1 to 10) for 33 different recipes from the dataset.

6.2.3. Measures

The metrics used are the same as in the previous test.

6.2.4. Subjects

The subjects were recruited in the same way as the previous evaluation, with 189 initial subjects restricted to 139 subjects (4587 ratings) by eliminating the subjects who did not provide the complete questionnaires. The dataset used is available from the authors upon request for academic purposes.

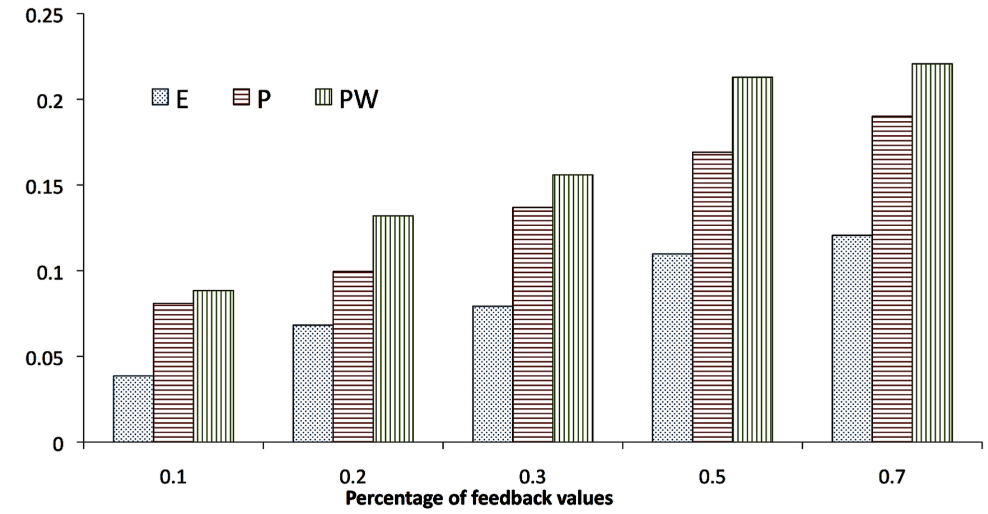
6.2.5. Accuracy Test

To verify the hypotheses H1 and H2, we assessed the performance of the three approaches in the way analogous to the one described for the events domain.

Figure 7 shows the average Spearman's rank correlation ρ for increasing percentage of feedback values in the training set. Again, in all the cases PW obtains the highest accuracy, followed

Copyright © 2015, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

Figure 7. Average ρ for different percentages of feedback values in the training set



by P and E. As before, PW obtains the highest percentage of users for $\rho \geq 0.5$ and the lowest percentage of user for $\rho < 0$. The difference between PW and E is significant for percentages of feedback values greater than 30% ($p = 0.023$).

The histogram in *Figure 8* shows the MAE values for the three algorithms for different percentage of feedback values in the training set, calculated as in the previous evaluation. When using MAE as evaluation metric, both P and PW perform better than the E approach, although in this case it seems that relevance of properties does not play such an important role as in the previous evaluation. This might be happening due to the fact that the property has ingredient usually has many values and therefore its relevance drops. Again, the disparity between algorithms grows with the percentage of feedback in the training set.

6.2.6. Diversity Test

To verify the hypothesis H3, we tested the diversity yielded by the three approaches analogously as in the previous diversity test, where we used the following nine categories: Meat, Seafood, Soup, Sauce, Rice, Vegetable, Pasta, Bread, Sweet (where we treated Mousse, Pancake, Cake and Cookie as Sweet).

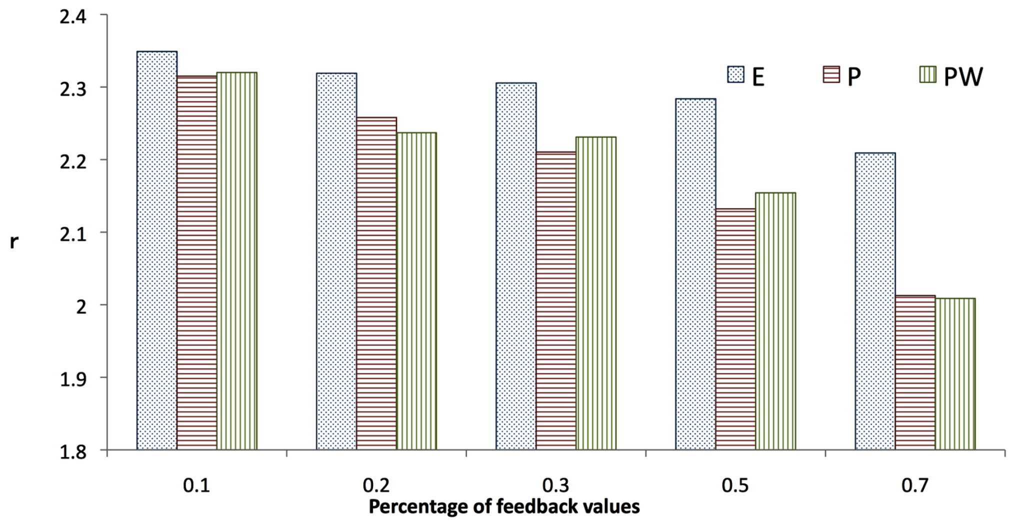
Figure 9 shows, for each algorithm, the average number of categories (vertical axis) represented in the set of the first n most similar objects for each domain object (horizontal axis).

We can see that for a low number of suggested items ($n < 5$), P and PW behave comparably, but already show better performance than algorithm E. For $n \geq 5$ it is clear that the best performing algorithm is PW, followed by P and E. For example, when suggesting 15 items, PW yields on average 3.9 categories, P yields 2.6 and E yields 1.4.

Also, PW behaves better than P and E w.r.t. diversity in this test, which means that the relevance assignment brings improvements in diversity calculation. This might be due to the fact that this technique assigned higher relevance to the certain properties, such as suitable for diet, which link items in different categories. The differences between the three methods are statistically significant, $p = 0.0007$ with Friedman test for $k > 2$ correlated samples. PW is different from both P ($p = 0.01$ with the Wilcoxon test) and E ($p = 0.005$). Hence, even without relatedness,

Copyright © 2015, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

Figure 8. Average MAE for different percentages of feedback values in the training set

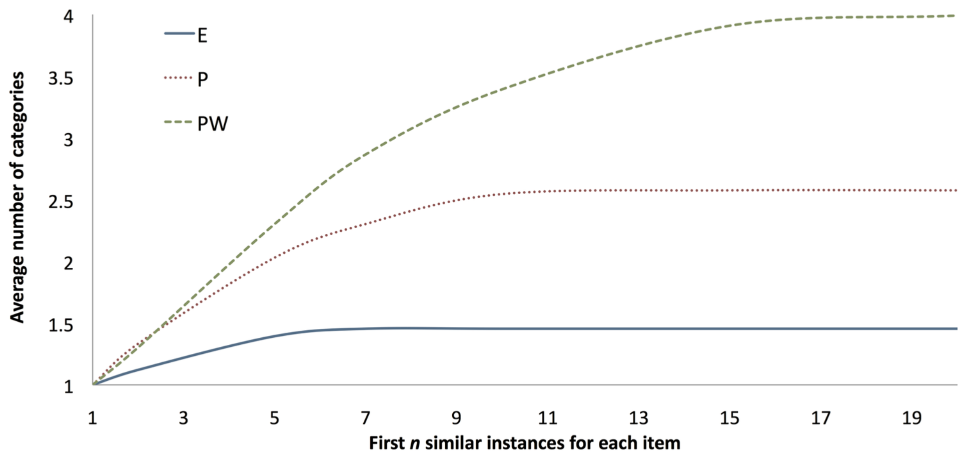


the propagation of interests based on property-based similarity offers satisfying results w.r.t. diversity of elements recommended to users.

6.2.7. Discussion

The second evaluation confirms that the property-based approaches outperform the baseline edge-based approaches, as far as accuracy and diversity are concerned. Using the Wikitaaable ontology did not allow us to perform the propagation using relatedness in addition to similarity, since the dataset did not contain the same values for different properties, apart from suitable for diet and not suitable for diet which are the properties with opposite meaning, hence could not

Figure 9. Diversity: average number of categories represented by the first n similar instances for each item



be used in relatedness calculation. But even with only property-based similarity, accuracy and diversity results are satisfying.

7. RELATED WORK

As could have been seen in the paper, our usage of ontologies is threefold:

- We represent the information in the user model using the domain ontology,
- We exploit the ontological structure to calculate similarity and relatedness among concepts,
- We use the ontological structure to propagate user interest values to similar and related objects.

Hence, in this section we describe the work in three areas of research. We first present some works that use ontologies to represent user models, followed by some relevant approaches in the literature for computing similarity among concepts, by some approaches for calculation of relatedness among concepts and finally, concluding with the approaches that propagate values over an ontology.

7.1. Ontological User Model

Our approach is built upon an *ontology-based user model*: we use an ontology to model the domain and we define the user model as an overlay over the domain ontology (Brusilovsky, 2007), annotating each concept with the value of user interest. Notice that we use the term "ontology-based user model" differently from other works, such as Heckmann et al. (2005), where it indicates an ad hoc ontology created to represent the user features in the user model (e.g. age, gender, profession, interest, knowledge). Other works in the literature using a similar ontology-based user model are the following ones.

Sieg et al. (2007) use ontological user profiles to improve personalized Web search. The user profiles are instances of a reference domain ontology, and they are incrementally updated based on the user interaction with the system.

In Middleton et al. (2004) the user feedback on research papers is collected and mapped onto the domain ontology representing the main concepts.

Razmerita et al. (2003) present OntobUM, an ontology-based user modeling for knowledge management systems. The domain ontology is annotated according to user interest in each concept.

Kalfoglou et al. (2001) represent user information associating a binary value of preference to each ontology concept, representing research themes. They then make use of ontological inference rules to propagate user preferences throughout the ontology and also across other different ontologies. The goal is, starting from a user query about a concept, to present the user with some related concepts.

Gauch et al. (2003) build an ontology-based user profiles starting from the Web pages the user visited. They compute similarity among such Web pages and the concepts in a domain ontology. Then, they annotate each concepts with this computed similarity value and add to the user model all the concepts with a value greater than zero.

Hatala and Wakkary (2005) present the museum guide ec(h)o, where the user interests are represented as a set of annotated concepts from the ontology of natural history. The concepts are annotated according to the strength of user interest, which is calculated based on the user activities in the physical museum place. The latest activities weigh more than past ones. Only concepts with value of interest greater than a certain threshold are present in the user model.

Copyright © 2015, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

7.2. Semantic Similarity

In addition to Tversky's *feature-based* model of similarity (Tversky, 1977) presented in this paper in Section 3, semantic similarity can be calculated in other two general modes: considering the information content or exploiting the ontology graph structure.

Resnik's notion of semantic similarity Resnik (1999) is based on *information content* in an is-a taxonomy, given by the negative logarithm of the probability of occurrence of the class in a text corpus. The information content of the closest class subsuming both compared concepts provides the shared information for both and gives the measure of their similarity.

Rada et al. (1989) use the ontology graph structure, exploiting the *distance* between nodes (i.e., the number of edges or the number of nodes between the two nodes) to calculate nodes similarity.

These three basic measures of similarity (feature-based, information content-based and distance-based) gave rise to many combined approaches. For example, Jiang and Conrath (1997) improve the distance based approach with the information content one. The strength of a link connecting two nodes is the difference between information content values of these two nodes. The weight of a link, in addition to link strength, accounts for local and average densities, depth of the parent node in the hierarchy and link type. Then the distance between two nodes is calculated as the shortest path linking the two nodes, using weighted links to traverse the paths and is used to calculate the similarity.

The semantic similarity introduced by Pirrò and Euzenat (2010) combines the feature-based model of similarity with the information theoretical one, where Tversky's function describing the saliency of the features is substituted by the information content of the concepts.

Lin (1998) introduces an information-theoretic definition of similarity based on a set of assumptions about similarity, calculated as the ratio between the amount of information that two concepts have in common and the amount of information needed to fully describe them. Lin's similarity measure is not tied to a particular knowledge representation and can be applied to any probabilistic model application.

Di Noia et al. (2012) use Linked Open Data (LOD) to develop a content-based movie recommender system in which they adapt a vector space model (VSM) approach to compute similarities between RDF resources. Similarly to us, they assume that two movies are similar if they have features in common. They represent the whole RDF graph as a 3-dimensional matrix where each slice refers to an ontology property and for each property the similarity between two movies is calculated using their cosine similarity. They also assign relevance factors to properties, either using a Genetic Algorithm or Collaborative-Filtering results derived from Amazon.

Hau et al. (2005) introduce a new semantic similarity measure for OWL objects based on the information theoretic measure of Lin (1998). They define the semantic similarity similarly to us, as a ratio between the shared and total information content of the two objects. The information content is defined using the objects' description sets which contain all the statements (triples) describing the given objects and their information content is based on their "inferencibility", i.e. the number of new RDF statements that can be generated by applying a certain set of inference rules to the predicate. They use their measure to determine the similarity of semantic services annotated with OWL ontologies.

In Ehrig et al. (2005) a new measure for similarity of ontological concepts is introduced. Their framework, based on abstract ontology model, consists of three layers for measuring similarity between entities: Data, Ontology and Context. In the Data layer the similarity of concepts is measured accounting for the data values of simple or complex data types. In the Ontology layer semantic relations between concepts are considered and edge-based similarity is used.

Copyright © 2015, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

Finally, the Context layer accounts for the ways concepts are used in some external contexts (similar concepts are used in similar contexts). These three layers are further enhanced with specific domain knowledge. The overall similarity is computed as a combination of these three individual similarity measures.

The similarity measure proposed by Bach and Dieng-Kuntz (2005) can be used to compare entities in two OWL DL ontologies. It is obtained as a combination of component similarity (based on the information extracted from the entity descriptions) and graph similarity.

Smyth (2007) takes into account individual features of concepts and each feature has its own similarity function defined for it. Similarly to us, they also introduce the weights which help distinguish the importance of individual features when calculating similarity.

Among many state-of-the-art similarity measures, we chose Tversky's feature-based model of similarity (Tversky, 1977) since we wanted to account for properties of domain objects in the similarity computation and Tversky's measure allows to include common as well as distinctive features of objects. In addition, Tversky's measure can be smoothly extended to calculate the relatedness among domain objects.

7.3. Semantic Relatedness

Karanastasi and Christodoulakis (2007) propose OntoNL Semantic Relatedness Measure used to calculate semantic relatedness between domain ontology concepts. Their relatedness measure combines three relatedness measures: relatedness based on properties (rel_{prop}), relatedness based on conceptual distance (rel_{CD}) and relatedness based on related senses (rel_{RS}). When calculating rel_{prop}, if the source concept is related to the target concept via an object property, their $rel_{prop} = 1$. Otherwise, they count the properties and the inverse properties the two classes have in common. We, on the other hand, count the properties with the same values which the classes have in common. rel_{CD} takes into account specificity factors based on the distance between the concepts and on the number of the object properties they have, in addition to edge distance between concepts. Finally, rel_{RS} counts the nouns and synonyms extracted from the descriptions of the concepts in the ontology that the concepts have in common.

Hirst and St-Onge (1998) consider Upward, Downward and Horizontal links and give 8 patterns for defining semantically correct paths. They give direction to each of the following relations: hypernymy, meronymy, hyponymy, holonymy, synonymy and antonymy. They then compute the semantic relatedness between concepts using the length of the shortest path between them considering all relations and patterns and the number of direction changes in the path. In their approach, each edge carries the same information content, i.e. has the same weight, presenting a problem.

Mazuel and Sabouret (2008) present a semantic relatedness measure on ontologies which considers object properties among concepts. The concepts can be connected following different kinds of relations. They first filter out the "semantically incorrect paths" using rules from Hirst and St-Onge (1998) and then assign a weight to each path depending on its type (is-a, part-of etc.), its context in the ontology and its position in the path. They use an information-theoretic approach to weight the hierarchical edges in the path, and a formula based on a "strength" of a given relation for non-hierarchical edges. In their work relatedness between two concepts is defined using the minimal weight of a semantically correct path between them and the maximal value of the distance introduced by Jiang and Conrath (1997).

Copyright © 2015, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

Gracia and Mena (2008) use the Web as knowledge source. Their semantic relatedness is based on the information about frequencies of term use, obtained from search engines. They generalize Normalized Google Distance (Cilibrasi and Vitanyi, 2007) to account for many different search engines. They propose the relatedness measure between words and extend it to ontology terms. Relatedness of ontology terms is based on their relatedness as synonyms and on the relatedness of their semantic descriptions, more precisely of their semantic contexts which characterize their meaning in the ontology (direct superclass for a class, domain classes for a property and the class the instance belongs to).

Witherell et al. (2009) propose to measure meronomic relatedness between concepts in an ontology, as a more generic measure of relatedness, which measures how much one concept is "part of" another. They introduce a weighted edge measure, in which each edge is assigned a weight based on the concept probability. The corpus for measuring concept probabilities is formed using multisets of concepts.

With respect to the relatedness measures above, we chose to exploit our method to calculate relatedness (Section 3) since it complements well the calculation of similarity (Section 3) and can be successfully used to propagate user interests in the ontology, thus enabling its evaluation to a certain extent. Our approach brings us one step closer to building the semantic relatedness measure applicable to wide range of real, complex ontologies. Since semantic relatedness can be used in the similar contexts as semantic similarity, it has a potential for application in various domains, such as word sense disambiguation (e.g. two synonym words would have different words related to them, based on their context of usage) or ontology matching (two corresponding concepts would be related to the same concepts).

7.4. Ontology-based Propagation

A few approaches exist that make use of propagation of values among ontology classes and instances.

Sieg et al. (2007, 2010) and Middleton et al. (2004) propagate the interest values in the ontology exploiting only is-a relationships in a fashion similar to Cena et al. (2011, 2013). They consider an ontology simply as a hierarchy of topics, where the topics can be used to classify items being recommended. Instead, in our current approach we make a complete use of the ontology (taxonomical structure, properties, instances) and of different forms of propagation.

In more details, Sieg et al. (2007, 2010) consider the ontology as as a semantic network and interest values are propagated using a spreading activation technique (Salton and Buckley, 1988). How much is propagated to each next classes depends on the weight of the relation. This value propagation takes place only in one direction (bottom-up, from instances to classes), whereas we perform a multi-directional propagation. In addition, they do not take into consideration the level of the node in the ontology where the propagation happens.

In Middleton et al. (2004), the is-a hierarchy is exploited to infer additional topics of interest (superclass topics starting from specific topics browsed by a user) for producing more complete profiles. The propagation of interest values in the taxonomy is based on is-a similarity. As opposed to us, their propagation is static and one-directional (bottom-up, from specific to generic concepts): given the interest value for a specific class, 50% of this value is spread to its super class.

Finally, Thiagarajan et al. (2008) represent user profiles as bags-of-words (BOW) where a weight is assigned to each term describing user interests. Then, they consider ontological relationships between the terms in BOW to propagate the values using graph spreading activation techniques. They use the ontology to find related terms.

Copyright © 2015, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

7.5. Recommender Systems based on Linked Open Data

The resources in the Web of Data (Bizer et al. (2009)) are expressed as LOD, Linked Open Data, i.e., described as RDF statements and by properties. In this sense, there is a certain similarity with our approach where the similarity and relatedness of domain concepts is also based on properties. A new generation of recommender systems is being developed taking advantage of this publicly available knowledge where the datasets are mapped onto their corresponding URIs which permit the extraction of the available properties for each resource in the dataset. A good selection of innovative linked open data-enabled recommender systems can be found in Di Noia et al. (2014).

W.r.t. similarity calculation, the approaches most similar to ours are: Peska and Vojtas (2014), where the grouping of equivalent features corresponds to our relatedness, Ampazis and Emmanouilidis (2014), where they exploit Tanimoto similarity, a variation of Tversky's feature similarity, and Heitmann and Hayes (2014), where constrained spreading activation is used.

8. CONCLUSION AND FUTURE WORK

In this paper we presented an approach for user interest propagation in an ontology, considering property-based similarity and relatedness of concepts and relevance of properties, in order to improve accuracy and diversity of the recommendation. We tested our algorithm using two datasets: one in the domain of event recommendation, the other in the domain of recipes, in order to assess the accuracy of recommendations derived from our property-based propagation (with similarity and relatedness as criterions for propagation) with respect to recommendations obtained by a state-of-the-art edge-based propagation. We also validated how adding relatedness to property-based propagation improves diversity in the results. Finally, we showed how our approach can be useful also for cross-domain recommendations (Fernández-Tobìas et al., 2012), i.e. for recommending items coming from different domains.

A prerequisite of our approach is a rich ontology with explicitly defined properties for classes. A simple taxonomy is not enough. Also, the calculation of similarity and relatedness for classes can be complex, but usually most of the user feedback in recommender systems is provided for instances, for which the calculation is pretty straightforward.

An important aspect in our approach is how to determine the importance of a property in order to weigh it in the propagation. In this paper, we inferred the relevance of a property as its capacity to determine the similarity between two entities for a set of users. However, we must notice that the importance of a property can change from users to users. For example, in a food domain, for one user the cost of food could be more important while for another user the calories would be important. Thus, as future work, we plan on automatic detection of the property importance for each specific user. To do so, we could find the most important properties for each user by making the intersection of the properties of the classes the user likes the most or by exploiting some machine learning approaches for automatic detection, such as the M5P algorithm.

Moreover, we want to combine property-based propagation with edge-based propagation of user interests presented in Cena et al. (2013) and to validate the whole approach and compare it with traditional spreading activation techniques.

Also, we would like to propagate the user interest also to the range of the property (class and the superclasses) of the object the user likes. For example, if the user likes the specific film The Ring, which has an object-property has genre with the class Horror as range, we will propagate the user interest also to the class Horror and to its superclasses (Action_Movie and Movie).

Copyright © 2015, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

As far as values for certain properties are concerned, it would be interesting to see how the similarity and relatedness of objects would change in the presence of hierarchical structure among these values.

Moreover, since we only consider object type properties in this work, dealing with data type properties, such as literals, might be an interesting area for future investigation. In the case, it would be necessary to determine when two literal values can be considered equal. For example, for string values, exact match could not be enough, since, for example, "New York", "City of New York" and "NY" should be equivalent.

Finally, we are currently experimenting our approach with Linked Open Data (LOD), thus using public LOD datasets in order to propagate user interest.

REFERENCES

Allemang, D., & Hendler, J. (2008). *Semantic Web for the Working Ontologist: Effective Modeling in RDFS and OWL*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.

Ampazis, N., & Emmanouilidis, T. (2014). Exploring semantic features for producing top-n recommendation lists from binary user feedback. In V. Presutti, M. Stankovic, E. Cambria, I. Cantador, A. Di Iorio, T. Di Noia, C. Lange, D. Reforgiato Recupero, & A. Tordai (Ed.), Semantic Web Evaluation Challenge, CCIS (Vol. 475, pp. 157–162). Springer International Publishing. doi:10.1007/978-3-319-12024-9_20

Antoniou, G., & van Harmelen, F. (2008). *A Semantic Web Primer* (2nd ed.). The MIT Press.

Bach, T.-L., & Dieng-Kuntz, R. (2005). Measuring Similarity of Elements in OWL DL Ontologies. *Proceedings of the AAAI-05 Workshop on Contexts and Ontologies: Theory, Practice and Applications.*

Beydoun, G., Lopez-Lorca, A. A., Garcìa-Sánchez, F., & Martìnez-Béjar, R. (2011). How do we measure and improve the quality of a hierarchical ontology? *Journal of Systems and Software*, *84*(12), 2363–2373. doi:10.1016/j.jss.2011.07.010

Bizer, C., Heath, T., & Berners-Lee, T. (2009). Linked Data - The Story So Far. *International Journal on Semantic Web and Information Systems*, *5*(3), 1–22. doi:10.4018/jswis.2009081901

Brusilovsky, P. (1996). Methods and techniques of adaptive hypermedia. *User Modeling and User-Adapted Interaction*, *6*(2-3), 87–129. doi:10.1007/BF00143964

Brusilovsky, P. (2007). Adaptive navigation support. In *The Adaptive Web, Methods and Strategies of Web Personalization*, *LNCS* (Vol. 4321, pp. 263–290). Springer.

Cena, F., Likavec, S., & Osborne, F. (2011). Propagating user interests in ontology- based user model. *Proceedings of the 12th International Conference of the Italian Association for Artificial Intelligence, AI*IA '11*, *LNCS* (V*ol. 6934,* pp. 299–311). Springer. doi:10.1007/978-3-642-23954-0_28

Cena, F., Likavec, S., & Osborne, F. (2012). Property-based interest propagation in ontology-based user model. *Proceedings of the 20th Conference on User Modeling, Adaptation, and Personalization, UMAP 2012*, *LNCS* (*Volume 7379*, pp. 38–50). Springer. doi:10.1007/978-3-642-31454-4_4

Cena, F., Likavec, S., & Osborne, F. (2013). Anisotropic propagation of user interests in ontology-based user models. *Information Sciences*, *250*, 40–60. doi:10.1016/j.ins.2013.07.006

Cilibrasi, R. L., & Vitanyi, P. M. B. (2007). The google similarity distance. *IEEE Transactions on Knowledge and Data Engineering*, *19*(3), 370–383. doi:10.1109/TKDE.2007.48

Cordier, A., Dufour-Lussier, V., Lieber, J., Nauer, E., Badra, F., Cojan, J., . . . Skaf-Molli, H. (2014). Taaable: A case-based system for personalized cooking. In Montani, S. & Jain, L. C. (Ed.), Successful Case-based Reasoning Applications - 2, volume 494 of Studies in Computational Intelligence, (pp. 121–162). Springer Berlin Heidelberg. doi:10.1007/978-3-642-38736-4_7

Copyright © 2015, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

Di Noia, T., Cantador, I., & Ostuni, V. C. (2014). Linked open data-enabled recommender systems: ESWC 2014 challenge on book recommendation. In V. Presutti, M. Stankovic, E. Cambria, I. Cantador, A. Di Iorio, T. Di Noia, C. Lange, D. Reforgiato Recupero, & A. Tordai (Ed.), Semantic Web Evaluation Challenge, of Communications in Computer and Information Science (Vol. 475, pp. 129–143). Springer International Publishing.

Di Noia, T., Mirizzi, R., Ostuni, V. C., Romito, D., & Zanker, M. (2012). Linked open data to support content-based recommender systems. *Proceedings of the 8th International Conference on Semantic Systems, I-SEMANTICS '12* (pp. 1–8). ACM. doi:10.1145/2362499.2362501

Ehrig, M., Haase, P., Hefke, M., & Stojanovic, N. (2005). Similarity for ontologies - a comprehensive framework. In Bartmann, D., Rajola, F., Kallinikos, J., Avison, D. E., Winter, R., Ein-Dor, P., Becker, J., Bodendorf, F., & Weinhardt, C. (Ed.), *Proceedings of the 13th European Conference on Information Systems, Information Systems in a Rapidly Changing Economy ECIS '05* (pp. 1509–1518).

Fernández -Tobìas., I., Cantador, I., Kaminskas, M., & Ricci, F. (2012). Cross-domain recommender systems: A survey of the state of the art. Proceedings of the 2nd Spanish Conference on Information Retrieval.

Gauch, S., Chaffee, J., & Pretschner, A. (2003). Ontology-based personalized search and browsing. *Web Intelligence and Agent Systems*, *1*, 1–3.

Gracia, J., & Mena, E. (2008). Web-based measure of semantic relatedness. *Proceedings of the 9th International Conference on Web Information Systems Engineering WISE '08* (pp. 136–150). Springer-Verlag. doi:10.1007/978-3-540-85481-4_12

Gruber, T. R. (1993). A translation approach to portable ontology specifications. *Knowledge Acquisition Journal*, *5*(2), 199–220. doi:10.1006/knac.1993.1008

Guarino, N., & Poli, R. (1995). Editorial: The role of formal ontology in the information technology. *International Journal of Human-Computer Studies*, *43*(5-6), 623–624. doi:10.1006/ijhc.1995.1064

Hatala, M., & Wakkary, R. (2005). Ontology-based user modeling in an augmented audio reality system for museums. *User Modeling and User-Adapted Interaction*, *15*(3-4), 339–380. doi:10.1007/s11257-005-2304-5

Hau, J., Lee, W., & Darlington, J. (2005). A semantic similarity measure for semantic web services. Proceedings of the Web Service Semantics Workshop at WWW 2005.

Heckmann, D., Schwartz, T., Brandherm, B., & Schmitz, M., & von Wilamowitz- Moellendorff, M. (2005). GUMO - the General User Model Ontology. *Proceedings of the 10th International Conference on User Modeling UM '05,* LNCS, (V*ol. 3538,* pp. 428–432). Springer.

Heitmann, B., & Hayes, C. (2014). Semstim at the lod-recsys 2014 challenge. In V. Presutti, M. Stankovic, E. Cambria, I. Cantador, A. Di Iorio, T. Di Noia, C. Lange, D. Reforgiato Recupero, & A. Tordai (Ed.), Semantic Web Evaluation Challenge, CCIS (Vol. 475, pp. 170–175). Springer International Publishing. doi:10.1007/978-3-319-12024-9_22

Hirst, G., & St-Onge, D. (1998). Lexical Chains as Representations of Context for the Detection and Correction of Malapropisms. In C. Fellbaum (Ed.), *WordNet: An Electronic Lexical Database* (pp. 305–332). MIT Press.

IJntema, W., Goossen, F., Frasincar, F., & Hogenboom, F. (2010). Ontology-based news recommendation. *Proceedings of the 2010 EDBT/ICDT Workshops*. ACM. doi:10.1145/1754239.1754257

Jiang, J., & Conrath, D. (1997). Semantic similarity based on corpus statistics and lexical taxonomy. *Proceedings of the International Conference on Research in Computational Linguistics* (pp. 19–33).

Kalfoglou, Y., Domingue, J., Motta, E., Vargas-Vera, M., & Shum, S. B. (2001). MyPlanet: an ontology-driven web-based personalised news service. *Proceedings of the Workshop on Ontologies and Information Sharing at IJCAI '01*.

Copyright © 2015, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

Karanastasi, A., & Christodoulakis, S. (2007). The OntoNL Semantic Relatedness Measure for OWL ontologies. *Proceedings of the 2nd International Conference on Digital Information Management ICDIM '07* (*Vol. 1*, pp. 333–338). doi:10.1109/ICDIM.2007.4444245

Kobsa, A., Koenemann, J., & Pohl, W. (2001). Personalized hypermedia presentation techniques for improving online customer relationship. *The Knowledge Engineering Review*, *16*(2), 111–155. doi:10.1017/S0269888901000108

Likavec, S. (2013). Shapes as property restrictions and property-based similarity. In O. Kutz, M. Bhatt, S. Borgo, & P. Santos (Eds.), *Proceedings of the 2nd Interdisciplinary Workshop The Shape of Things* (V*ol. 1007,* pp. 95–105).

Lin, D. (1998). An information-theoretic definition of similarity. *Proceedings of the 15th International Conference on Machine Learning ICML* '98 (pp. 296–304). Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

Mazuel, L., & Sabouret, N. (2008). Semantic relatedness measure using object properties in an ontology. *Proceedings of the 7th International Conference on The Semantic Web, ISWC '08* (pp. 681–694). Springer. doi:10.1007/978-3-540-88564-1_43

Middleton, S. E., Shadbolt, N. R., & De Roure, D. C. (2004). Ontological user profiling in recommender systems. *ACM Transactions on Information Systems*, *22*(1), 54–88. doi:10.1145/963770.963773

Peska, L., & Vojtas, P. (2014). Hybrid recommending exploiting multiple dbpedia language editions. In V. Presutti, M. Stankovic, E. Cambria, I. Cantador, A. Di Iorio, T. Di Noia, C. Lange, D. Reforgiato Recupero, & A. Tordai (Ed.), Semantic Web Evaluation Challenge, of Communications in Computer and Information Science (Vol. 475, pp. 144–149). Springer International Publishing. doi:10.1007/978-3-319-12024-9_18

Pirroò, G., & Euzenat, J. (2010). A feature and information theoretic framework for semantic similarity and relatedness. *Proceedings of the 9th International Semantic Web Conference ISWC '10,* LNCS (V*ol. 6496,* pp. 615–630). Springer. doi:10.1007/978-3-642-17746-0_39

Rada, R., Mili, H., Bicknell, E., & Blettner, M. (1989). Development and application of a metric on semantic nets. *IEEE Transactions on Systems, Man, and Cybernetics*, *19*(1), 17–30. doi:10.1109/21.24528

Razmerita, L., Angehrn, A., & Maedche, A. (2003). Ontology-based user modeling for knowledge management systems. *Proceedings of the 9th International Conference on User modeling, UM '03,* LNCS (V*ol. 2702,* pp. 213–217). Springer. doi:10.1007/3-540-44963-9_29

Resnik, P. (1995). Using information content to evaluate semantic similarity in a taxonomy. In Chris S. Mellish (Ed.), *Proceedings of the 14th International Joint Conference on Artificial Intelligence, IJCAI '95,* (*Vol. 1*, pp. 448– 453). San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.

Resnik, P. (1999). Semantic similarity in a taxonomy: An information-based measure and its application to problems of ambiguity in natural language. *Journal of Artificial Intelligence Research*, *11*, 95–130.

Salton, G., & Buckley, C. (1988). On the use of spreading activation methods in automatic information. *Proceedings of the 11th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval SIGIR '88* (pp. 147–160). ACM. doi:10.1145/62437.62447

Salton, G., & McGill, M. (1984). *Introduction to Modern Information Retrieval*. McGraw-Hill Book Company.

Sarwar, B., Karypis, G., Konstan, J., & Riedl, J. (2001). Item-based collaborative filtering recommendation algorithms. *Proceedings of the 10th International Conference on World Wide Web WWW '01* (pp. 285–295). New York, NY, USA. ACM.

Sieg, A., Mobasher, B., & Burke, R. (2007). Web search personalization with ontological user profiles. *Proceedings of the 16th ACM Conference on Information and Knowledge Management CIKM '07* (pp. 525–534). ACM. doi:10.1145/1321440.1321515

Copyright © 2015, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

Sieg, A., Mobasher, B., & Burke, R. (2010). Improving the effectiveness of collaborative recommendation with ontology-based user profiles. *Proceedings of the 1st International Workshop on Information Heterogeneity and Fusion in Recommender Systems HetRec '10* (pp. 39–46). ACM. doi:10.1145/1869446.1869452

Smyth, B. (2007). Case-based recommendation. In *The Adaptive Web, Methods and Strategies of Web Personalization, LNCS* (Vol. 4321, pp. 342–376). Springer.

Sullivan, D. O., Smyth, B., & Wilson, D. (2004, March). O'Sullivan, D., Smyth, B., & Wilson, D. C. (2004). Preserving recommender accuracy and diversity in sparse datasets. *International Journal of Artificial Intelligence Tools*, *13*(1), 219–235. doi:10.1142/S0218213004001491

Thiagarajan, R., Manjunath, G., & Stumptner, M. (2008). Finding experts by semantic matching of user profiles. *Proceedings of the 3rd Expert Finder Workshop on Personal Identification and Collaborations: Knowledge Mediation and Extraction PICKME '08*.

Tversky, A. (1977). Features of similarity. *Psychological Review*, *84*(4), 327–352. http://wikitaaable.loria.fr/rdf/ doi:10.1037/0033-295X.84.4.327

Witherell, P., Krishnamurty, S., Grosse, I., & Wileden, J. (2009). A meronomic relatedness measure for domain ontologies using concept probability and multiset theory. *Proceedings of the Annual Meeting of the North American Fuzzy Information Processing Society NAFIPS '09* (pp. 1–6). IEEE. doi:10.1109/NAFIPS.2009.5156444

Silvia Likavec is a post-doctoral researcher at Computer Science Department, University of Torino, Italy. She obtained her PhD in Computer Science from University of Torino, Italy and École Normale Supérieure de Lyon, France in 2005 and her Master in Applied Mathematics from University of Novi Sad, Serbia in 2005. Her research activity is focused on the following: (i) modelling the user in social systems, in particular the design of the ontology-based user model and the propagation of user interests using this model; (ii) semantic similarity among ontological concepts; (iii) theoretical computer science, in particular type assignment systems, resource control calculi and semantics of programming languages. She is the author of more than 40 articles in prestigious international journals, monographs and peer-reviewed conferences and workshops and two text books for university level courses.

Francesco Osborne is a research assistant at the Knowledge Media institute (KMi) of the Open University in Milton Keynes, UK. His research focuses on Semantic Web, Semantic Publishing, User Modelling and Data Mining. Currently he is working on data mining algorithms for extracting knowledge from academic data. He is the main developer of Rexplore, a system which leverages novel solutions in large-scale data mining, semantic technologies and visual analytics, to provide an innovative environment for exploring and making sense of scholarly data.

Federica Cena is an assistant professor at the Department of Computer Science at the University of Torino (Italy). She is a member of the SIOS, Smart Interactive Objects Systems group at the Department of Computer Science. She received her PhD in 2007 in the area of semantic web for distributed user-adaptive applications. She is working on user modeling, personalization and ubiquitous computing, Web 2.0 and Semantic Web for user modeling and adaptation. In the last years, she is mainly devoted to studying the implications of Internet of Things for user modeling and personalisation. She has served as a program committee member of several international conferences and workshops. She has organized several international workshops on user modeling.

Copyright © 2015, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.