

Inferring Semantic Relations by User Feedback

Francesco Osborne, Enrico Motta

Knowledge Media Institute, The Open University, MK7 6AA, Milton Keynes, UK
{francesco.osborne, e.motta}@open.ac.uk

Abstract. In the last ten years, ontology-based recommender systems have been shown to be effective tools for predicting user preferences and suggesting items. There are however some issues associated with the ontologies adopted by these approaches, such as: 1) their crafting is not a cheap process, being time consuming and calling for specialist expertise; 2) they may not represent accurately the viewpoint of the targeted user community; 3) they tend to provide rather static models, which fail to keep track of evolving user perspectives. To address these issues, we propose Klink UM, an approach for extracting emergent semantics from user feedbacks, with the aim of tailoring the ontology to the users and improving the recommendations accuracy. Klink UM uses statistical and machine learning techniques for finding hierarchical and similarity relationships between keywords associated with rated items and can be used for: 1) building a conceptual taxonomy from scratch, 2) enriching and correcting an existing ontology, 3) providing a numerical estimate of the intensity of semantic relationships according to the users. The evaluation shows that Klink UM performs well with respect to handcrafted ontologies and can significantly increase the accuracy of suggestions in content-based recommender systems.

Keywords: Ontology, User Modelling, Recommender Systems, Ontology-based User Modelling, Data Mining, Ontology Learning, Community-based Ontologies.

1 Introduction

In the last ten years, ontology-based recommender systems have been shown to be effective tools for predicting user preferences and suggesting items. Many of them [1,2,3,4] build user models as overlays of the domain ontology and use variations of the spreading activation technique for propagating the user feedback on certain items to related concepts. This solution allows recommender systems to suggest items that are semantically similar to the ones that the user liked and to compare users according to their preferences on a variety of concepts. In most cases, the ontologies used by these methods are manually crafted in OWL, both to facilitate sharing and because this language enjoys good tool support.

There are however some issues associated with the ontologies adopted by these approaches, such as: 1) their crafting is not a cheap process, being time consuming and calling for specialist expertise; 2) they may not represent accurately the viewpoint of the targeted user community; 3) they tend to provide rather static models, which fail to keep track of evolving user perspectives.

A common way to craft these ontologies is to consult domain experts, who however, may disagree on how to represent the different semantic relationships or may propose solutions that, while describing a correct formalization of the domain, may not be the most adequate for a recommender system. For example, users may take decisions on the basis of features that were instead neglected in the expert crafted ontology. Of course, it is possible to evaluate a first draft of the ontology on a sample of users and then iterate the crafting process; however this is a time consuming and expensive process. Moreover, the final product is a static knowledge base that will eventually need to undergo new modifications, e.g., when adding new categories of items to the recommender system.

For all these reasons, a more appealing perspective is to consider the domain ontology, and in particular the semantic relationships between concepts, as something dynamic that can be learned, adjusted and adapted according to the emergent semantics that characterize a group of users. The idea of deriving community-based ontologies from social networks or folksonomies has been investigated by a number of authors, yielding promising results [5,6]. A possible drawback of these ontologies is that they usually strongly depend on the community taken in consideration. However, this actually becomes an advantage when the aim is to adapt an ontology to that same community. Adapting ontologies to specific users is also the idea which gave origin to personal ontology views [7] (POVs), which proved to be effective tools in assisting tasks like web navigation and search, allowing the users to classify items according to their own mental categories [8].

We thus propose to combine these two ideas (extracting ontologies from communities and tailoring an ontology to particular users) in the context of recommender systems by exploiting user ratings for eliciting emergent semantics and then adapting the ontology to these users for improving the recommendations accuracy. The flow of information thus becomes bidirectional: the user preferences are used to adapt and enrich the domain ontology and the ontology is exploited to infer additional user preferences.

Ontologies are formal specifications of a shared conceptualization and thus they should theoretically express “a shared view between several parties, a consensus rather than an individual view” [9]. Hence, the pretension of tailoring them on a particular group of users or on a specific aim, such as recommending items, may indeed appear preposterous. However, not even ontologies escape the popular George Box paradigm stating that “all models are wrong, but some are useful” [10]. In this spirit, we want to be able to select among the possible ontologies describing a certain domain the one which works best in forecasting the preferences of a specific group of users, by exploiting state of the art algorithms for propagating user preferences in ontology-based recommender systems, such as those presented in [1,2,4]. Hence, we do not claim that an ontology crafted or enriched by means of user feedback would necessarily be the most complete or formally correct representation of a domain: only that it will work better than the available alternatives for that specific task.

As an example, by analysing user ratings we may detect that users who like the Italian cheese “Gorgonzola” tend to like also “Blue Danish” more often than one might expect on the basis of their actual semantic relationships: in fact in that ontology they may simply be two subclasses of “Cheese”, with no property in common. Hence, this situation can be addressed by analysing these two types of products, discovering that they are both blue mould cheeses and add either a common

superclass, “Blue Mould Cheese”, or a related property. In the same way, we may also learn the intensity of the different semantic relationships according to the users. For example, we may discover that the relationship between “Wine” and “White Wine” is stronger than the one between “Juice” and “Orange Juice”, even if both relationships are *subClassOf*. We can then use this knowledge to compute a more accurate semantic distance between concepts and thus foster the recommendation process.

As a contribution to addressing this issue, we propose *Klink UM* (Klink for User Modelling), an algorithm which generates semantic relationships between concepts using as input the user ratings on items associated with keywords. Klink UM is a modified version of *Klink* [11], an algorithm designed by the authors of this paper to mine semantic relationships between research areas. Klink was developed for Rexplore [12], a novel tool that provides a variety of functionalities and visualizations to support users in exploring information about the academic domain. Klink UM uses similar statistical and machine learning techniques for finding hierarchical and similarity relationships between keywords associated with rated items and can be used for: i) building a conceptual taxonomy from scratch, ii) enriching and correcting an existing ontology, iii) providing a numerical estimate of the intensity of the semantic relationships according to a group of users.

The rest of the paper is organized as follows. In section 2, we describe the Klink UM algorithm, focusing in particular on the changes with respect to the original Klink algorithm. In section 3 we evaluate the approach i) by comparing the generated taxonomies with two gold standard human crafted ontologies and ii) by applying Klink UM to a content-based recommender system with the aim of increasing the accuracy of recommendations. Section 4 deals with the related work. In section 5 we summarize the main conclusions and outline future directions of research.

2 The Klink UM Algorithm

2.1 Overview of the Approach

Most ontology-based recommender systems rely mainly on the conceptual taxonomy defined by semantic relationships such as *subClassOf* [1,2,4]. Klink UM can be used to infer both hierarchical and similarity relationships and adopts by default the SKOS model¹, a standard way to represent knowledge organization systems using RDF. In SKOS it is possible to express a taxonomy by stating that a concept is more or less specific than another. Thus, the hierarchical links detected by Klink UM (see section 2.3) are mapped to *skos:broaderGeneric*, a property from the *SKOS 5 model*, which indicates that a concept is broader than another. For example, “Music” is broader than “Rock Music”. Similarly, strong similarity links between concepts (see section 2.3) are mapped to the *relatedEquivalent* relationship, which we define as a sub-property of *skos:related*, to indicate that two particular ways of referring to a concept can be treated as equivalent. A trivial case is when there are lexical variations of the same tag, e.g., “rock-music” and “Rock Music”.

¹ <http://www.w3.org/2004/02/skos/>

Klink UM can be used in two modalities: i) to build a conceptual taxonomy and ii) to enrich, correct and/or give suggestions for improving an existing ontology. In the first case the input is a collection of user ratings associated with keywords, tags or categories and the result is an OWL model and a matrix associating each relationship with an intensity score. In the second case, the input includes also the original ontology and the output yields the enriched ontology, the intensity matrix and, when possible, some suggestions for further modifications.

When feeding an ontology to Klink UM, it is also possible to associate a weight to each semantic relationship. The higher the weight, the more resilient to changes will be the relationship. The given ontology is treated as a taxonomy shaped by hierarchical links whose strength is defined by the weights. The links will be included in the set of hierarchical links discovered by Klink UM (section 2.3) and may be deleted if stronger links are found (section 2.5). It is however possible to preserve a relationship despite any counter-evidence by assigning a weight equal to infinity.

The approach herein presented includes several new features with respect to the (original) Klink algorithm. Among them: 1) the possibility of using user ratings as input, 2) the ability of examining and correcting an existing ontology and 3) the capacity of suggesting changes to an ontology or signalling discrepancies between the ontology and the user feedback.

Pseudocode 1 – The KlinkUM Algorithm

```

function KlinkUM (RATINGS, KEYWORDS, OWL, OWL_weights) returns
(NEW_OWL, NEW_OWL_weights) {
  RATINGS = a set of user ratings on the keywords/tags/categories;
  KEYWORDS = a set of keywords/tags/categories;
  OWL = a initial OWL Ontology, optional;
  OWL_weight = a set of weights associated with the ontology relationships, optional;
  con_prob = computeConditionalProbabilities(RATINGS); // Step 1
  keywords_to_merge=true;
  while (keywords_to_merge) {
    foreach K in KEYWORDS {
      co_keywords = selectKeywordsWithRatingsInCommon(K);
      foreach K2 in co_keywords { // Step 2
        linkH = computeHL(K, K2, con_prob, RATINGS);
        if (linkH >  $t_h$ ) links["H", K, K2]= linkH; // hierarchical link
        else {
          linkS = computeSL(K, K2, RATINGS);
          if (linkS >  $t_{ss}$ ) links["S", K, K2]= linkS; // strong similarity link
          else if (linkS >  $t_{ws}$ ) links["WS", K, K2] = linkS; // weak similarity link
        }
      }
    }
    links = filterKeywords(KEYWORDS, links); // Step 3
    if (at least one weak similarity link in links)
      clusters = clusterSimilarityLinks(links); // Step 4
    if (at least one strong similarity link in links)
      KEYWORDS = mergeKeywords(links, KEYWORDS);
    else keywords_to_merge=false;
  }
}

```

```

links = fixLoops(links); // Step 5
links = enforceStructuralRequirements(links, OWL_weight);
OWL_NEW = OWL; // Step 6
foreach (link or concept discrepancy between OWL and links )
    NEW_OWL = proposeModification(discrepancy);
foreach (cluster in clusters) NEW_OWL = missingPropertyOrSuperClass(cluster);
NEW_OWL_weights = normalizeWeights(links);
return NEW_OWL, NEW_OWL_weights;
}

```

The steps of the algorithm are the followings:

- 1) The matrix representing user ratings on the keywords is used for computing the conditional probability that a user who has given a positive or negative feedback on keyword x would give the same feedback on keyword y .
- 2) Each keyword is compared with the other keywords with which it shares at least n ratings in common in order to infer the *hierarchical links*, which shape the conceptual taxonomy, and the *strong/weak similarity links*, which denote the degree of similarity between keywords.
- 3) The keywords are filtered and tidied up and those that do not relate to other keywords or appear to be outside the target domain are excluded;
- 4) The keywords that share a strong similarity link are merged together, and the keywords that share a weak one are clustered together. Steps 2-4 are repeated with the new keywords obtained by merging the keywords with inferred equivalence relationships, until no new *similarity link* is inferred.
- 5) The links are tidied up by deleting loops and redundancies; the user's requirements on the structure are enforced;
- 6) If an initial ontology was given, a series of suggested *modifications* with respect to it and some *alerts* about possibly missing properties or super concepts may be proposed to the user. The algorithm returns an OWL file and a matrix yielding the detected intensity of hierarchical and similarity relationships.

We will now explain more in detail how the individual steps are carried out.

2.2 Step 1 – From Ratings to Conditional Probability

Klink UM relies on variations of the subsumption model [13,14] for detecting hierarchical links. The subsumption model is used for finding hierarchical relationships between terms associated with documents. Term x is said to subsume term y if two conditions holds: $P(x/y) = 1$ and $P(y/x) < 1$, e.g., if y is associated to documents that are a subset of the documents x is associated to. Usually the first condition is relaxed in $P(x/y) > \alpha$, since it is quite improbable to find a perfect relationship, with $0.7 < \alpha < 0.8$.

As discussed in [11], Klink originally computed the conditional probability of keyword x given keyword y by using the ratio of the co-citations to the total citations of y . Since Klink UM considers ratings instead than co-citations, it calculates the conditional probability that a user who has a positive or negative opinion on x will have the same opinion on y . This is computed as the ratio between common positive/negative feedbacks and the total positive/negative feedbacks received by a

keyword. A rating from a user above/below her/his average rating by a chosen threshold constitutes a positive/negative feedback on a keyword. Let us consider the case in which a user rates 7 the keyword “Beer”, 8 the keyword “Wine” and has an average rating of 6.5. If we choose a threshold for the difference equal to 1, Wine has a positive feedback, but not so Beer. With a threshold equal to 0.5 both receive a positive feedback. We call this a *common positive feedback*, since it relates to the same user. Thus for a common positive feedback the difference between the given rating and the average rating of the user must be positive and higher than a threshold for both keywords. The *common negative feedback* follows the same rule with the difference that in this case the difference must be negative. Even if in many systems a user is not allowed to rate directly the keywords, the rating of a keyword can be estimated by using the average rating of the items associated with it.

For example, if keyword A received 50 feedbacks and 25 of them were in common with keyword B , the conditional probability of the feedbacks $P_f(B|A)$ is equal to 0.5, indicating a very strong relationship between the two keywords. To have a better idea about the direction of the subsumption relationship, we need to compute also $P_f(A|B)$: for example if $P_f(A|B)=0.1$ we have a good evidence that A may be a sub-concept of B , since many people who like A also like B , whereas only a limited number of people who like B are into A . However, if $P_f(A|B)=0.5$ we will still be clueless about the direction of the relationship: A and B might be similar concepts or even synonymous.

2.3 Step 2 – Inferring Hierarchical and Similarity Relationships

In this section we will elaborate on inferring the hierarchical and similarity links between keywords. We will use the first kind of link to build the conceptual taxonomy, and the second one to merge together keywords that point to a single concept and to suggest relationships between concepts that may not be explicit in the initial ontology.

Inferring Hierarchical Links. A hierarchical link of keyword x with respect to y is inferred when the difference between the conditional probabilities $P_f(y|x)$ and $P_f(x|y)$ is high enough and the two terms are considered fairly similar by the users. More formally, we compute the strength of the hierarchical relationship as:

$$L(x,y) = \left(\frac{P_f(y|x)}{\log(D_x)} - \frac{P_f(x|y)}{\log(D_y)} \right) \cdot \cos(\hat{x}, \hat{y}) \cdot (1 + \text{sim}(x,y)) \quad (1)$$

where $\cos(\hat{x}, \hat{y})$ is the cosine similarity between the two user ratings vectors; $\log(D_i)$ is the logarithm of the number of items associated with keyword i ; $\text{sim}(x,y)$ is the percentage of common characters between x and y with respect to the longer keyword. The number of items associated with a keyword is needed to balance the cases, not so uncommon during the cold start phase, in which a relatively smaller keyword may have received a higher number of feedbacks than its super-concept. This may bias the sample and reverse the link direction.

A hierarchical link is inferred when $L(x,y) > t_h$ and then x is considered a candidate for becoming a sub-concept of y . The value of $L(x,y)$ will be also used to weight the intensity of a semantic relationship.

Inferring Similarity Links. The similarity between two keywords x and y is computed according to the following formula:

$$S(x, y) = \frac{\cos(\hat{x}, \hat{y})}{\max(\cos_{sup}(\hat{x}, \hat{y}), \cos_{sib}(\hat{x}, \hat{y}))} \quad (2)$$

where $\cos_{sup}(\hat{x}, \hat{y})$ and $\cos_{sib}(\hat{x}, \hat{y})$ are the average cosine similarities with the common super-concepts and the sibling concepts. The last ones are the sub-concepts of the same super-concepts, according to the detected hierarchical relationships. Hence, this formula does not only check that two keywords are generally similar, but also that they are more similar to each other than they are with their siblings and super-concepts. This is important since it is normal for related concepts to be quite similar, especially if they are in the lower levels of a conceptual taxonomy.

Using this formula we infer two kinds of links: the strong similarity link and the weak one. The first correspond to $S(x, y) > t_{ss}$, the second to $S(x, y) > t_{ws}$, where $t_{ss} > t_{ws}$.

The strong similarity link is used for the identification of synonymous or related keywords that point to the same concept. The weak similarity link is utilized for the detection of clusters of similar keywords that may indicate the presence of an implicit super-concept or propriety, not reflected by the current ontology.

Estimating the threshold values. Assigning a sound value to t_h , t_{ss} , and t_{ws} is important for generating a conceptual taxonomy that is optimized for inferring user preferences. While is possible to assign these values empirically and vary them according to the desired sensibility as we did in [11], in most case it is better to rely on an automatic method. Hence, we use the Nelder-Mead algorithm [15], which is a derivative-free optimization method, used to solve parameter estimation problems when the function values are uncertain. It considers the parameters to be found as vertices of a simplex, which is a generalization of the notion of a tetrahedron to arbitrary dimensions. Then it performs a sequence of geometrical transformations on it, aimed at minimising an evaluation function.

In this case we need a function that measures the ability of the ontology in yielding sound suggestions to the users. Here, we adopt as evaluation function the Spearman's rank correlation coefficient ρ (see section 3.2) computed between the lists of items suggested using spreading activation [3] on 50% of the rated items and the list produced by ordering the other half according to their ratings. This procedure was also used in [3,4] for evaluating the accuracy of ontology-based recommender systems.

2.4 Step 3 and 4 – Keyword Filtering and Merging

To filter out keywords that are just noise or are related to not relevant domains, Klink UM applies mainly three techniques: 1) it deletes keywords without inferred relationships with any other keyword; 2) it uses the common feedback distributions to detect and delete keywords that are too general; 3) it uses external knowledge from web pages about a domain to check the estimated dimension of the keywords in that same domain and then deletes those under a certain threshold. These methods are also used in Klink, and discussed more thoroughly in [11].

The keywords which share a strong similarity link are consider synonymous, thus they will be merged together and at the next iteration of the algorithm they will be

considered as a single keyword with a rating vector given by the average of the rating vectors of the merged keywords. The keywords that share a weak link will be clustered together, but they will preserve their individuality. The cluster will be used to generate the alert relative to potential discrepancies between the original ontology and the perspective of the users. In fact the clustered keywords point to a situation that should be recognized also in the ontology, for example by adding a common super-concept or a shared property. Both merging and clusterization are implemented by means of a bottom-up single-linkage hierarchical clustering algorithm which uses the inverse of $S(x,y)$ as the distance between the keywords.

The algorithm will then return to step 2 if new similarity links are inferred in this iteration, otherwise it will proceed to step 5.

2.5 Step 5 – Tidying up the Keywords and Adjusting the Links

The links are reassessed by detecting the loops and breaking them up by eliminating the weaker links in terms of $L(x,y)$. Redundant links are also deleted. A redundant link is a link that is unnecessary because implicit in other relationships: for example if A is a sub-concept of B and B a sub-concept of C, we do not need to state explicitly that A is a sub-concept of C.

This phase includes the enforcement of the user requirements on the structure. At the moment Klink UM supports two main structure boundaries, which are the maximum number of super and sub concepts. They are implemented by deleting the links in excess with lower $L(x,y)$ score. As anticipated in section 2.2, a semantic relationship included in the initial ontology with an assigned weight w can be deleted only for inserting links with $L(x,y) > w$.

2.6 Step 6 – Suggestions and OWL Creation

If the algorithm did not receive an ontology in input, it outputs an OWL model and the matrix containing detected intensities of the semantic relationships. The intensity scores of the relationships (equal to $L(x,y)$) can be used to weight the links of the conceptual taxonomy and enhance a variety of approaches [1,2,4] that rely on graph-based distance to assess semantic similarity between concepts.

As stated before, Klink UM produces the OWL by mapping the hierarchical links to the *skos:broaderGeneric* semantic relationships and the strong similarity links to the *relatedEquivalent* relationships. However it is up to each individual implementation to decide whether to use the default SKOS-based model or to produce instead an alternative representation of the hierarchical structure.

If an input ontology is given, the algorithm generates a list of suggestions that can be answered with a yes or no by the user. For each detected discrepancy between the given ontology and the generated one, the algorithm suggests a modification to the original ontology, e.g., adding a new *skos:broaderGeneric* relationship between two previously unrelated concepts. At the moment Klink UM can suggest: 1) to add a relationship, 2) to delete a relationship, 3) to add a concept, 4) to delete a concept. After the user validates the suggestions, the algorithm proceed to generate a new OWL model. Of course the user can also decide to trust Klink UM and accept all suggestions by default.

At the end, Klink UM will also yield a warning about potentially neglected properties linking the component of the clusters found via the *weak similarity links*. In this case, Klink UM does not try to implement any automatic modification, and only reports potential problems that an ontology engineer may want to address.

3 Evaluation

In this section we aim to prove that 1) Klink UM can generate conceptual taxonomies similar enough to the ones crafted by human experts and 2) the ontologies generated or enriched by Klink UM are tailored to a particular group of users, and thus particularly useful for recommendation purposes.

Hence, in the first part we will measure the F-measure between conceptual taxonomies generated by Klink and gold standard expert crafted ontologies. In the second part we will compare the accuracy of the suggestions yielded by a content-based recommendation system when using a human crafted ontology, the same ontology enriched by Klink UM, and an automatically generated conceptual taxonomy.

3.1 Ontology generation

In order to evaluate the ability of Klink UM to generate a conceptual taxonomy from scratch we used two ontologies, designed about two years ago by experts in the gastronomic domain and ontology engineers for an adaptive application called WantEat [16], developed as part of the PIEMONTE Project. WantEat is an application for Android and iPhone that allows a user to explore the “slow food” domain. The users can give a feedback by tagging, voting, visiting and bookmarking both items and categories. In this case, items are gastronomic products, such as a particular Parmesan cheese sold by a certain producer, while categories include general concepts, such as “Parmesan Cheese”, “Fat Cheese” and “Cheese”.

The two ontologies are 1) *Cold Cuts*, a three level ontology with 19 classes, describing the different cuts of meat and 2) *Drinks*, a three level ontology with 33 classes, describing different kinds of drinks. Our hypothesis is that Klink UM should be able to generate OWL ontologies that are very similar to the human crafted ones by analysing user ratings on the concepts included in the ontology. This approach was tested against the classic subsumption method as in [13] and [14], using the conditional probability that the average user who likes/dislikes keyword x will have the same relationship with keyword y , as described in section 2.2.

We used the dataset collected for [4] which includes user ratings on cuts of meat (in particular cold cuts) and on drinks obtained by mean of online questionnaires. The ratings ranged between 0 and 10 and the threshold for the negative/positive feedback described in Section 2.2 was set to 1. The initial sample for the *Cold Cuts* included 1392 ratings given by 87 subjects, 19-45 years old, recruited according to an availability sampling strategy. The sample for the *Drinks* ontology included 7623 ratings given by 231 subjects, in the age range 19-38 years old, similarly recruited.

We ran Klink UM and the baseline method 10 times for each different set of randomized input data and compared the generated ontologies with the two original

gold standard ontologies, using the average recall, precision and F-measure (that is their harmonic mean) of the inferred relationships.

Figure 1 shows the F-measure of the two approaches with respect to the *Cold Cuts* and the *Drinks* ontologies. Clearly, in both cases Klink UM performs better than the subsumption method, with the two resulting curves showing a statistically significant difference ($p < 10^{-12}$, according to the chi-square test). Klink UM is able to obtain at the largest sample size a Precision of 96% with a Recall of 94% for *Drinks* ($N = 7623$) and a Precision of 87% with a Recall of 80% for *Cold Cuts* ($N = 1392$).

The performance of Klink UM depends on two factors: 1) the fraction of keywords voted by the average user (μ) and 2) the number of ratings. The first component is important since Klink UM needs to compare the votes of the same user on different keywords in order to infer the hierarchical links: if these data are too sparse, this becomes difficult. The left panel of Figure 2 shows the Klink UM performance on both *Drinks* and *Cold Cuts* as a function of μ . It can be seen that it performs well for both ontologies, with the *Drinks* dataset yielding better results thanks to its higher number of ratings.

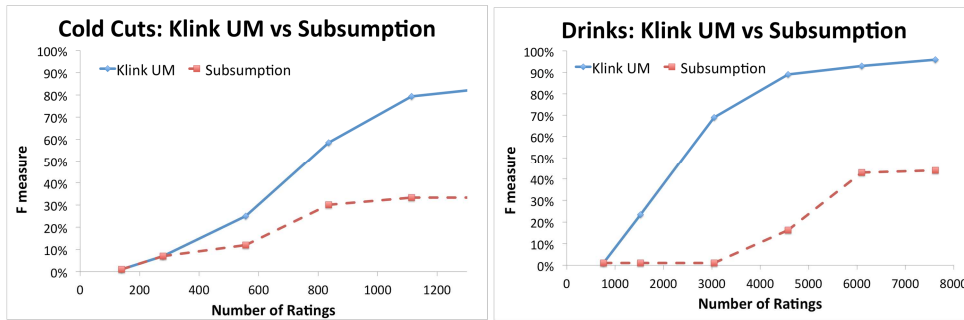


Figure 1. F-measure of Klink UM and the Subsumption method for the *Cold Cuts* and the *Drinks* datasets.

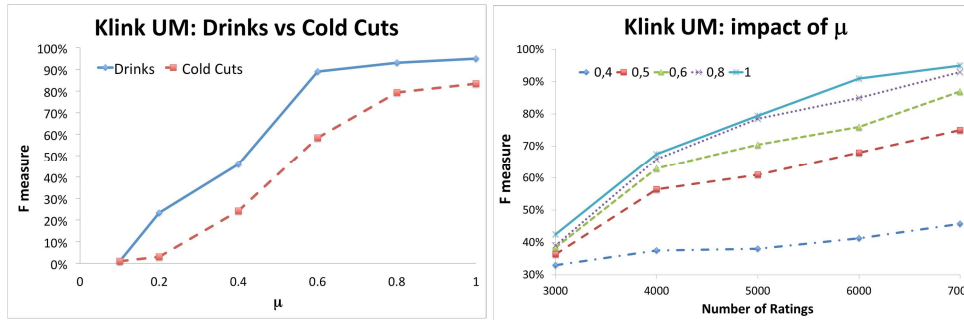


Figure 2. On the left: the performance of Klink UM in the two tests as a function of μ . On the right: the trade-off between μ and ratings for the *Drinks* dataset.

The right panel of Figure 2 highlights the trade-off relationship between the number of ratings and the μ value for the *Drinks* dataset. If μ is high enough, Klink UM is able to obtain very good results even with a low number of ratings: with $\mu = 0.8$, Klink UM is able to reach an F-measure of 75% with only 5000 ratings, whereas with $\mu = 0.5$ it needs 7000 ratings to reach the same F-measure.

It is interesting to notice that the curves exhibit a progressively increasing crowding with the increasing value of μ ; the gap between the curves corresponding to $\mu = 0.4$ and $\mu = 0.6$ is ten times larger than the gap between $\mu = 0.8$ and $\mu = 1$. The chi-square test confirms this behaviour: the probability that the difference between the $\mu = 0.4$ and $\mu = 0.6$ curves may be ascribed to chance is $p < 10^{-12}$, increasing to $p = 2 \times 10^{-7}$ for the $\mu = 0.6$ and $\mu = 0.8$ curves, and finally losing statistical significance with $p = 0.93$ for the $\mu = 0.8$ and $\mu = 1$ curves.

Figure 3 shows a portion of the version of the *Drinks* ontology generated by Klink UM, highlighting the intensity of the subsumption relationships according to the users. For example, it appears that “Spumante” (the Italian version of Champagne) is considered a less typical “Wine” than “Red Wine” and “White Wine”. Thus if we want the ontology to mirror this perception we should differentiate “Spumante” from its siblings “Red Wine” and “White Wine” by adding a property or by using a different super-concept for “Spumante”. Of course a group of users with different background and drinking habits may have a different idea on this subject.

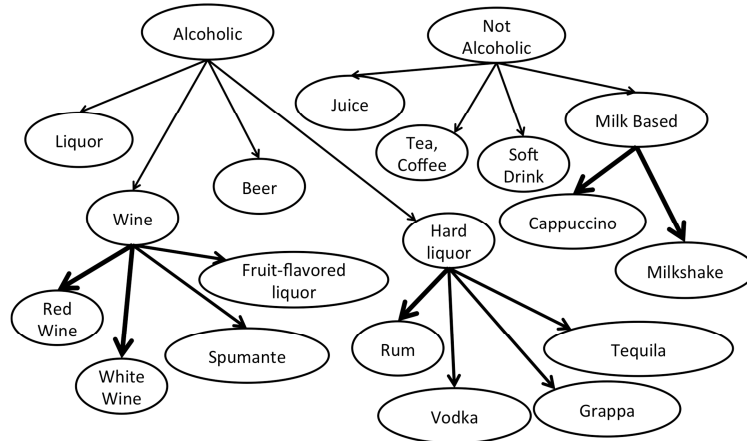


Figure 3. A portion of the *Drinks* ontology generated by Klink UM with 5000 user ratings. The width of a link is proportional to the detected intensity of the semantic relationship.

The placement of fruit-flavoured liquor under Wine is formally a mistake since accordingly to the human crafted ontology it should be under Hard Liquor. However by looking at the ratings we can see in this case a stronger correlation with the Wine concept. As the number of ratings increase this may be revealed as a statistical fluctuation or rather it may confirm that our users considered it more similar to the Wine concept. Hence, the ontology used for recommendation purpose may be modified accordingly, e.g., by adding a common property.

3.2 Ontology Enrichment and Generation for Content-based Recommender Systems

Many state of the art approaches use ontologies or conceptual taxonomies for inferring additional user interests from an initial set of ratings and then suggesting items. A standard technique is to use spreading activation to propagate user interests

from a set of initial concepts or items to the semantically related concepts. To measure the ability of Klink UM in assisting the recommendation process we will use the approach described in Cena et al [4], which was shown to outperform other similar techniques, such as [1,2]. The links between concepts were weighted by the intensity detected by Klink UM (see formula 1), when available.

In particular we will compare the accuracy of three approaches, namely:

- Spreading activation on an expert crafted ontology (labelled **S**)
- Spreading activation on an expert crafted ontology, corrected and enriched by accepting by default Klink UM suggestions (labelled **SE**)
- Spreading activation on a conceptual taxonomy generated from scratch by Klink UM (labelled **SG**)

To compute the accuracy we rely on the *Drinks* dataset described in the previous section. The accuracy of a certain approach was measured by giving to it only a certain fraction r of user ratings and then comparing the produced recommendations with the true user preferences. The comparison was done using Spearman's rank correlation coefficient ρ , which provides a non-parametric measure of statistical dependence between two ordinal variables and gives an estimate of the relationship between two variables using a monotonic function.

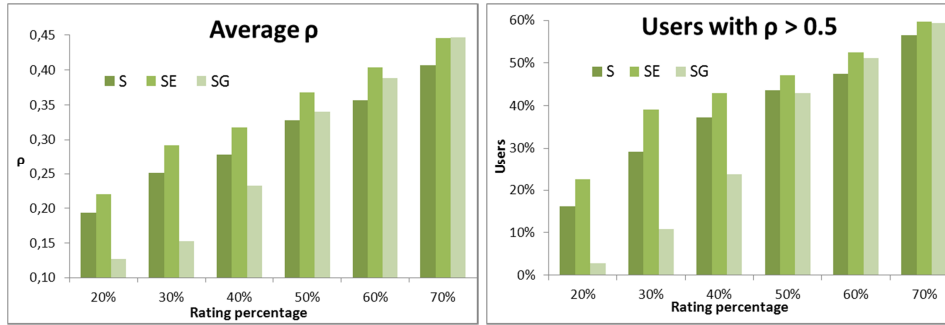


Figure 4. Average ρ (left panel) and number of users with $\rho > 0.5$ (right panel), when taking as input a certain rating percentage r for the three techniques.

Figure 4 shows the performance of the three approaches for different percentages of input ratings. SE always outperforms S, and is significantly different from it for $r \leq 30\%$ ($0.002 \leq p \leq 0.026$, according to the chi-square test). In fact, as highlighted by the right panel of the figure 4, when $r \leq 30\%$, SE obtains on the average 8.1% more user with $\rho > 0.5$ than S, while for $40\% \leq r \leq 70\%$, the difference is reduced to 4.3%. Hence, especially in situations of data sparsity, when the system does not yet know much about user preferences, Klink UM is able to significantly improve the quality of the recommendation by enriching the initial ontology.

The SG algorithms, which tries to learn the conceptual taxonomy from scratch, does not perform as well as S for $r \leq 40\%$. However, for higher values of r SG is not significantly different from S and SE ($0.67 \leq p \leq 0.98$) and for $r \geq 60\%$ the performance of SE and S are almost identical, both of them being superior to S. Hence, while it takes a decent amount of user feedback to learn the conceptual taxonomy from scratch, once this is achieved, the results are indistinguishable from the version that relies on the expert crafted ontology. Hence, SG seems a viable option especially for systems that can rely on a good number of user ratings and for

which the manual crafting of the domain ontology is not easy. In all other cases the best solution appears to start with a human crafted ontology and then to enrich and correct it accordingly to the user needs.

4 Related Work

In the first part of this section, we will describe the state of the art in techniques to infer conceptual taxonomies or semantic relationships. In the second part we will highlight the main works relative to ontology-based recommender systems, which can benefit from Klink UM.

The idea of extracting ontologies from user communities is thoroughly discussed in the work of Mika [5], which extends the traditional bipartite model of ontologies with the social dimension, proposing a tripartite model of actor, concepts and instances. Similarly, Specia et al [6] extract semantics from folksonomies by clustering tag sets and detecting highly related tags corresponding to concepts in ontologies. The automatic inference of semantic relationships is usually addressed by means of two approaches. The first was developed in the area of computational linguistic and exploits lexico-syntactic patterns [17], the second uses clustering techniques [18]. The Lexico-Syntactic Pattern Extraction (LSPE) is an approach which discovers relationships between terms by exploiting patterns like “such as...”, “and other...”, and so on. For example, De Cea et al [19] use this technique to infer ontological relationships, such as *subClassOf*. Instead the approaches that rely on clustering techniques build a hierarchy of keywords according to a variety of similarity metrics. For example, in [20] a hierarchical clusterization algorithm is applied to the context of web pages and a top-down partitioning is used to generate a multi-way-tree taxonomy from the binary tree. The *TaxGen* framework [21] uses instead a hierarchical agglomerative clustering algorithm and text mining techniques for building a taxonomy from a set of documents. Also Klink UM uses a hierarchical algorithm and similarity distances between keywords, but only for the inference of the similarity links and for the detection of potentially missing superclasses or properties.

The subsumption approach, exploited also by Klink UM, was introduced in Sanderson and Croft [13]. Also Schmitz et al [14] use a subsumption-based model for inducing a faceted ontology from *Flickr* tags. The metric we propose for finding hierarchical links exploits the same idea, but considers also the reciprocal conditional probability and other factors, such as the cosine similarity between keywords. The subsumption approach inspired also the GrowBag algorithm [22], which uses a biased *PageRank* algorithm to exploit second order co-occurrences.

While Klink UM aims to adapt an ontology to a groups of users, other approaches tailor ontologies to specific users, resulting in personal ontology views [7]. For example, Haase et al [8] proposed a method for assisting users in the management of their personal ontologies with the aim of yielding more accurate recommendations.

Klink UM can be useful especially for ontology-based recommenders [1,2,3,4], since it makes it easier to craft and update an ontology targeted to a group of users. It can currently identify only hierarchical and similarity relationships, however most works in the fields also rely solely on these relationships. For example Middleton et al [1] exploit the user feedback on research papers and use the hierarchical relationships between classes to infer other topics of interest. In Sieg et al [2] the ontology is

treated as a semantic network and the interest values are updated by means of spreading activation. Cena et al [4] propose instead a multi-directional anisotropic interest propagation which is able to spread user feedback also to instances.

Many other methods exploit the ontology graph structure to compute the distance between concepts. For example, Resnik et al presented a semantic similarity measure [23] based on information content in a taxonomy that is computed as the negative logarithm of the probability of occurrence of the class in a text *corpus*. Similar metrics are also applied to determine the similarity between Linked Data entities [24]. Other methods, such as [3], use instead shared and distinctive OWL properties rather than a graph-based distance. We believe that Klink UM can be helpful to all these approaches as a support for computing a fit-for-purpose conceptual similarity between concepts.

5 Conclusions

In this work we presented Klink UM, an extension of the Klink algorithm which is able to detect relationships between keywords and create or enrich an ontology starting from a set of user ratings on the keywords, with the aim of tailoring the ontology to a specific group of users.

We tested the ability of Klink UM to build a conceptual taxonomy from scratch and to assist the recommendation process. In the first task it overperformed the subsumption approach obtaining an F-measure of 95% for the *Drinks* test (N= 7623) and of 83% for the *Cold Cuts* test (N=1392). In the second one, the approach relying on an ontology enriched by Klink UM outperformed the one relying on the human crafted one, especially in conditions of data sparsity ($p \leq 0.03$). Moreover, after a good number of user ratings, the conceptual taxonomy crafted by Klink UM performed as well as the human crafted enriched ontology ($p \geq 0.67$).

Klink UM can also be used for generating suggestions about potential missing properties, that may have been forgotten or considered irrelevant when the ontology was crafted. Hence, it allows ontology engineers and domain experts to gain an interesting user-centred prospective.

The next step will be to have Klink UM recognizing groups of people with different views of the domain in order to build different version of the domain ontology, tailored to them [25]. We also are working on novel heuristics for detecting a higher number of semantic relationships.

References

1. Middleton, S. E., Shadbolt, N. R. and De Roure, D. C.: Ontological user profiling in recommender systems. *ACM Transactions on Information Systems*. (2004)
2. Sieg, A., Mobasher, B., Burke, R.: Web search personalization with ontological user profiles. In *Proceeding of the 16th ACM Conference on Information and Knowledge Management. CIKM '07*. ACM, pp. 525–534. (2007)
3. Cena, F., Likavec, S., Osborne, F.: Property-based interest propagation in ontology-based user model. In *Proceeding of UMAP 2012*. Montreal, Canada. (2012)
4. Cena, F., Likavec, S., Osborne, F.: Anisotropic propagation of user interests in ontology-based user models. *Information Sciences*. (2013)

5. Mika, P.: Ontologies are us: A unified model of social networks and semantics. In *The Semantic Web–ISWC 2005* (pp. 522-536). Springer Berlin Heidelberg. (2005)
6. Specia, L., Motta, E.: Integrating folksonomies with the semantic web. In *The semantic web: research and applications* (pp. 624-639). Springer Berlin Heidelberg. (2007)
7. Chaffee, J., Gauch, S.: Personal ontologies for web navigation. In *Proceedings of CIKM 2000*. (2000)
8. Haase, P., Hotho, A., Schmidt-Thieme, L., Sure, Y.: Collaborative and Usage-driven Evolution of Personal Ontologies. In *Proceedings of 2nd European Semantic Web Conference, Heraklion, Greece*. Springer. (2005)
9. Guarino, N., Oberle, D., & Staab, S.: What is an Ontology? In *Handbook on ontologies* (pp. 1-17). Springer Berlin Heidelberg. (2009)
10. Box, G. E. P., and Draper, N. R., (1987), *Empirical Model Building and Response Surfaces*, John Wiley & Sons, New York, NY.
11. Osborne, F., Motta, E.: Mining Semantic Relations between Research Areas. In: *Proceedings of the 11th International Semantic Web Conference*. Boston, USA. (2012)
12. Osborne, F., Motta, E., Mulholland, P.: Exploring Scholarly Data with Rexplore. In *The Semantic Web–ISWC 2013* (pp. 460-477). Springer Berlin Heidelberg. (2013)
13. Sanderson, M., Croft, B.: Deriving concept hierarchies from text. In: *Proceedings of the SIGIR conference*, pp. 206–213. (1999)
14. Schmitz, P.: Inducing Ontology from Flickr Tags. In *Proceedings of the 15th International Conference on World Wide Web (WWW)*. Edinburgh, UK. (2006)
15. Olsson, D. M., Nelson, L. S.: The Nelder-Mead simplex procedure for function minimization. *Technometrics*, 17(1), 45-51. (1975)
16. Console, L. and Piemonte Team: WantEat: interacting with social networks of intelligent things and people in the world of enogastronomy. In *Proceedings of the Workshop on Interacting with Smart Objects*, pp. 1–6. (2011)
17. Morin, E.: Automatic acquisition of semantic relations between terms from technical corpora. In *Proceedings of the 5th International Congress on Terminology and Knowledge Engineering*. (1999)
18. Assadi, H.: Construction of a regional ontology from text and its use within a documentary system. In N. Guarino (ed.), *Formal Ontology in Information Systems*, *Proceedings of FOIS-98*, pp. 236-249. Trento, Italy. (1999)
19. De Cea, G., de Mon, I., Montiel-Ponsoda, E.: From Linguistic Patterns to Ontology Structures. In *Proceeding of the 8th Conference on Terminology and Artificial Intelligence*. (2009)
20. Chuang, S., Chien, L.: A practical web-based approach to generating topic hierarchy for text segments. In *Proceedings of the 13th ACM Conference on Information and Knowledge Management*. Washington, D.C., USA. (2004)
21. Müller, A., Dorre, J.: The TaxGen Framework: Automating the Generation of a Taxonomy for a Large Document Collection. In *Proceedings of the 32nd Hawaii International Conference on System Sciences-Volume 2*, pp. 20-34. (1999)
22. Diederich, J., Balke, W., Thaden, U.: Demonstrating the Semantic GrowBag: Automatically Creating Topic Facets for FacetedDBLP. In: *Proceedings of JCDL '07*, ACM, New York, NY, USA. (2007)
23. Resnik, P.: 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. (1999)
24. Di Noia, T., Mirizzi, R., Claudio Ostuni, V., Romito, D. and Zanker, M.: Linked Open Data to support Content-based Recommender Systems. In *Proceeding of the International Conference on Semantic Systems (I-Semantics)*. (2012)
25. Osborne, F. A POV-Based User Model: From Learning Preferences to Learning Personal Ontologies. In *User Modeling, Adaptation, and Personalization* (pp. 376-379). Springer Berlin Heidelberg. (2013)