Vincent Schickel-Zuber[1]
Boi Faltings[1]

[1] Artificial Intelligence Lab (LIA) liawww.epfl.ch
EPFL, Station 14, 1015 Lausanne, Switzerland
{vincent.schickel-zuber,boi-faltings}@epfl.ch

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

Inferring User's Preferences using Ontologies

1. Recommendation Problem

Recommendation Problem: How to help the user find the right item on the internet?

A.) Collaborative Filtering (CF): Recommends item to the user based on the experience of like-minded groups of other users

Works in 4 steps:
1. Rating elicitation - ask the user to rate a set of items.
2. Representation – represents the users' rating and items in a user-item matrix.
3. Neighborhood Formation – starts by computing the similarity between {users/items} using a proximity measure and then finds the list of the k most similar {users/items}.
4. Generate Recommendation: predict items based on what the k-most similar {users/items} liked.

Advantages & Disadvantages:
- ✓ Satisfactory performance when sufficient data is available & low cognitive requirement on the users.
- ✗ Profound and well known problems: cold-start, first-rater, sparsity, privacy and scalability.

B.) Preference-Based: Recommends items based on the user's own preferences

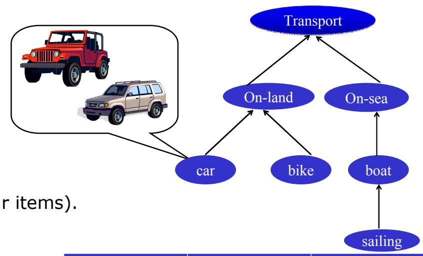
Works in 3 steps:
1. Elicit the weights and utility functions.
2. Build a personal user model based on the elicited preferences.
3. Generate Recommendation: predict items based on the user model.

Advantages & Disadvantages :
- ✓ Is theoretically 100% accurate & overcome almost all of CF's problems (except cold start)
- ✗ Preference elicitation is a major problem for many reasons (privacy, elicitation overload,…)

2. Using an Ontology to model the user's preferences

What is an ontology? It's an explicit specification of a conceptualization of the real *world*
Example of real world ontologies: WordNet, GeneOntology, Amazon.com Taxonomies.
Formally for our task: A Directed Acyclic Graph (DAG) where a node represents primitive concepts and an edge models a binary specialization relation (ISA).
➡ A concept represents a group of items with the same *features*.

The recommendation Problem can be seen as the problem of predicting a **score *S*** assigned to an item



What is a score? A real valued function [0..1] that shows how much a user likes a concept (set of similar items).
Based on three fundamental assumptions:
- **A1**: The score depends on the feature of the concept.
- **A2**: Each feature contributes independently to the score.
- **A3**: Unknown and disliked features make no contribution (implies a lower bound model).

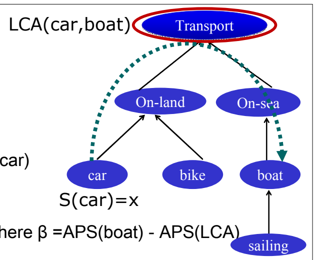
Computing an A-priori Score (APS) of a concept. Leverages semantic information in ontology structure.
APS models the expected score of each concept for an average user, using **only** the structure of the ontology.

How do we compute the APS?
- Without any user information, suppose the score is uniformly distributed => $P(S(c)>x)=1-x$
- Concept has n descendants, and under assumption A3, $P(S(c)\leq x)=1-(1-x)^{n+1}$
- $E(S(c))=\int xf(x)dx = 1 / (n+2)$ ➡ **APS(c) = 1 / (n+2)** where n is the number of descendants of c.

Concept	#descendants	APS
Car	0	½
Bike	0	½
Sailing	0	½
Boat	1	⅓
On-land	2	¼
On-sea	2	¼
Transport	6	⅛

3. Using the Ontology to infer missing knowledge

How to infer missing knowledge (score)? Lets S(*car*) = x, how to estimate S(*boat*)?
1. Find the lowest common ancestor to the concept *car* and *boat*, LCA(car, boat), which is the concept *transport*.
2. Infer the knowledge upwards from the concept *car* to LCA(car, boat).
3. Finally, infer the knowledge downwards from LCA(car, boat) to the destination concept *boat*.



A.) Upward Inference S(LCA|car): Infer the score of the LCA from the concept *car*
When going up, we are removing known features => score decreases
Assumption A1 => score is distributed over its features
} $S(LCA|car)=\alpha S(car)$ where $\alpha=APS(LCA) / APS(car)$

B.) Downward Inference S(boat|LCA): Infer the score of the concept *boat* from the ancestor LCA
When going down, we are adding unknown features to the ancestor => score increases
Assumption A2 => feature contributes independently to the score
} $S(boat|LCA)=S(LCA) + \beta$ where $\beta = APS(boat) - APS(LCA)$

C.) Upward & Downward Inference S(boat|car): Infer the score of any concept through the lowest common ancestor LCA
There is a chain from concept car to the concept boat (but not a path)
As for Bayesian network, assume independence between upward and downward path
A.) Upward Inference + B.) Downward Inference
} $S(boat|car)=\alpha S(car) + \beta$

4. Application of the Model (1)

Derive a new similarity metric (OSS) from the inference model
➡ **$D(x,y) = -\log(\alpha)+\log(1+\beta)$**

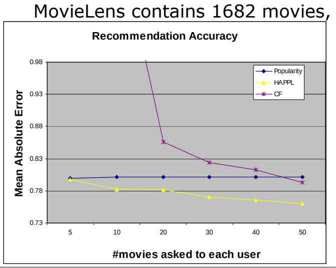
Tested the similarity metric on the WordNet ontology version 2.0.

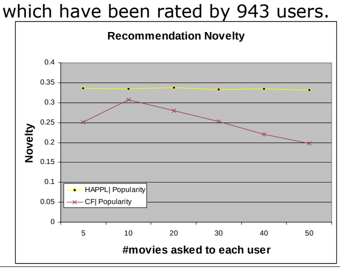
The experiment was as follows[2].
1. Took 28 word pairs that have human ratings associated to it
   The word pairs covers high, medium, and low level of similarity
2. Use different metrics to evaluate the similarity of these pairs
   Metrics: Edge Based[2], Resnik[2], Jiang[3], OSS.
3. Measure the correlation between the metrics and human ratings

Metric	Edge Based	Resnik	Jiang	OSS
Correlation	0.603	0.793	0.859	**0.908**

5. Application of the Model (2)

Derived a recommendation system from our model (HAPPL)
Built a movie recommendation systems using the MovieLens data set
MovieLens contains 1682 movies, which have been rated by 943 users.





References: [2] Resnik, P. 1998. Semantic Similarity in a Taxonomy: An Information-Based Measure and its Application to Problems of Ambiguity in Natural Language, Journal JAIR, (11) pp 95-130; [3] Jiang, J., and Conrath, D. 1998. Semantic Similarity based on corpus and lexical taxonomy, Proc. Of 10th Cof. COLING.