Integrating Tags in a Semantic Content-based Recommender

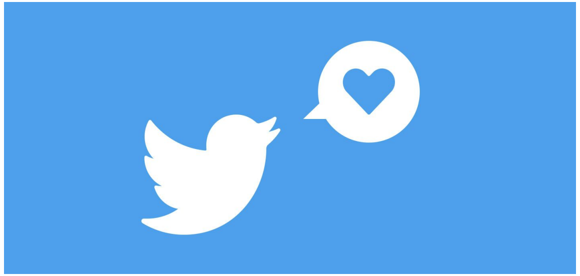
Jae Yong Kim

Degemmis, Marco & Lops, Pasquale & Semeraro, Giovanni & Basile, Pierpaolo. (2008). Integrating tags in a semantic content-based recommender. RecSys'08: Proceedings of the 2008 ACM Conference on Recommender Systems. 163-170. 10.1145/1454008.1454036.

Contents

Introduction

- User roles: Recipient → Participant
- folksonomy => folks + taxonomy == #Tags
  - Socially constructed classification schema
- Does the integration of tags cause an increase of the prediction accuracy in the process of recommending items to users?



Recommender System

1. Content Analyzer
    a. Semantic Indexing
2. Profile Learner
    a. Multivariate Poisson Model
3. Recommender
    a. ITR (ITem Recommender)

        + UGC(User Generated Contents)

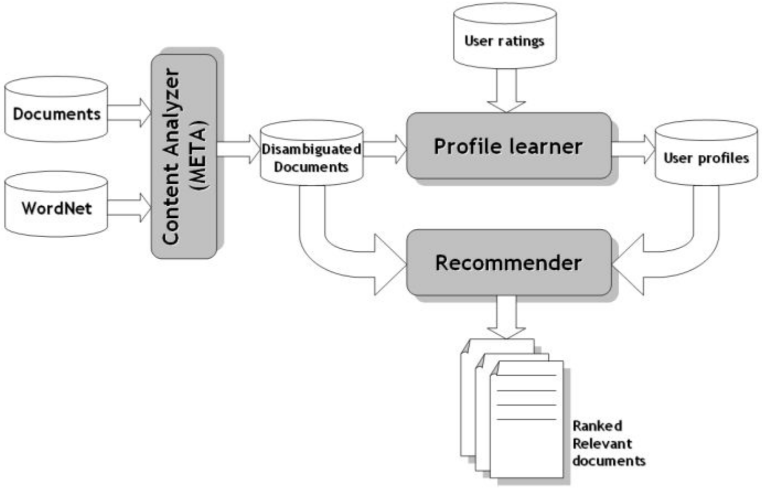


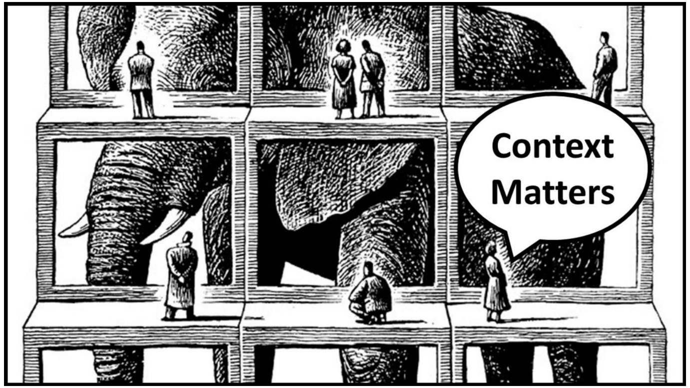
Figure 1: ITR architecture

Content Analyzer

- Documents: Textual description of items

Relevant ***Concepts*** surrounding the content(vs. keywords?)

- final output: Disambiguated document
- How?
    - repository for word senses ⇒ WORDNET 2.0
    - Word Sense Disambiguation (WSD)



WSD - JIGSAW

Determining which word is right in the situation

d = $[w_1, w_2, \ldots, w_h]$

Semantic similarity = relatedness between word (A, B) -->Leacock-Chodorow measure

$$\mathrm{sim}_{\mathrm{LC}}(c_1, c_2) = -\log \frac{\mathrm{len}(c_1, c_2)}{2 \times \max_{c \in WordNet} \mathrm{depth}(c)}$$

JIGSAW(d) = WORDNET synsets (Synonym-set)X = $[s_1, s_2, \ldots, s_k]$ $(k \leq h)$

BOS (Bag-of-Synsets)

** BOS: Synset-based vector space representation

Textual Slots : item property representation

$$d_n^s = \langle t_{n1}^s, t_{n2}^s, \ldots, t_{nD_{ns}}^s \rangle \xrightarrow{\text{rep in vector space}} f_n^s = \langle w_{n1}^s, w_{n2}^s, \ldots, w_{nD_{ns}}^s \rangle$$

s = index of slot

n = nth document in N-documents

t = set of all different synsets found in slot

w = weight of synset

(frequency of synset tn)

Learning User Profile

Multivariate Bernouli vs Multinomial Model

Problems

1. Variation in length of documents
2. Rare categories (Not enough samples)

Let's use the Poisson distribution(model) for learning the bayes text classifier !

$$
\begin{aligned}
P(c|d_j) &= \frac{P(d_j|c)P(c)}{P(d_j|c)P(c) + P(d_j|\bar{c})P(\bar{c})} \\
&= \frac{\frac{P(d_j|c)}{P(d_j|\bar{c})}P(c)}{\frac{P(d_j|c)}{P(d_j|\bar{c})}P(c) + P(\bar{c})}
\end{aligned} \tag{1}
$$

If we set:

$$
z_{jc} = log \frac{P(d_j|c)}{P(d_j|\bar{c})} \tag{2}
$$

then Eq. (1) can be rewritten as:

$$
P(c|d_j) = \frac{e^{z_{jc}}P(c)}{e^{z_{jc}}P(c) + P(\bar{c})} \tag{3}
$$

Multivariate Poisson Model

$$z_{jc} = \sum_{i=1}^{|V|} w_{ij} \cdot log \frac{\lambda_{ic}}{\mu_{ic}}$$

$$\lambda_{ic} = \frac{\#\text{occurrences for } t_i \text{ in the pos. training documents}}{\#\text{total tokens in the pos. training documents}},$$

$$\mu_{ic} = \frac{\#\text{occurrences for } t_i \text{ in the neg. training documents}}{\#\text{total tokens in the neg. training documents}}.$$

V: Vocabulary size

w: frequency term of t in document d

$$z_{jc}^s = \sum_{i=1}^{|V|} w_{ij}^s \cdot log \frac{\lambda_{ic}^s}{\mu_{ic}^s}$$

$$\lambda_{ic}^s = \frac{1}{|D_c|} \sum_{j=1}^{|D_c|} \hat{w}_{ij}^s \qquad \mu_{ic}^s = \frac{1}{|D_{\bar{c}}|} \sum_{j=1}^{|D_{\bar{c}}|} \hat{w}_{ij}^s \qquad s = 1, \ldots, M \tag{10}$$

where D_c ($D_{\bar{c}}$) is the number of documents in class c ($\bar{c}$),

$$\hat{w}_{ij}^s = \frac{w_{ij}^s}{\alpha \cdot avgtf^s + (1 - \alpha) \cdot avgtf_j^s} \tag{11}$$

Training

- User has some discrete scale (MIN and MAX)
- positive training set if ratings > (MIN + MAX)/ 2
- negative training set if ratings < ((MIN + MAX)/ 2
- compute a-posteriori classification scores $P(c+|d_j)$ and $P(c-|d_j)$, given new document d_j

Augmenting Recommender

ITR += static documents + dynamic user generated content(tags)

Tags => SocialTags(I), PersonalTags(U, I), PersonalTags(U)

1. WSD (JIGSAW): **Use static content as context instead of other tags
2. Profile learner : infers the profile as a binary text classifier
3. a-priori probabilities of profile_like and profile_dislike

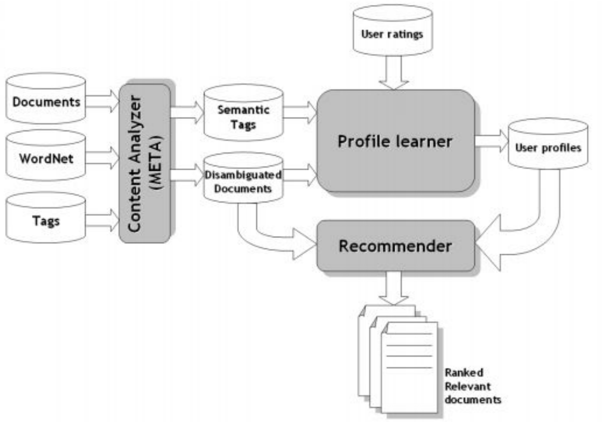


Figure 2: ITR 2.0 architecture

Experiment / Datasets

45 paintings chosen from the collection of the Vatican picture-gallery

- title, artist, description + tags and preference score on 5 points scale (1 = strongly dislike, 5= strongly like)

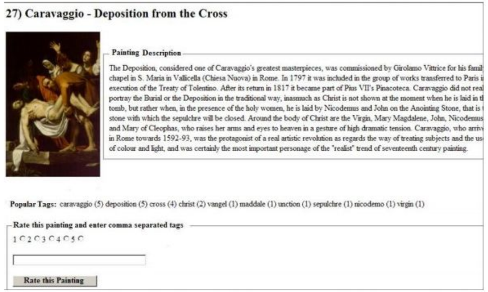
#1 only static

#2 only SemanticPersonal

#3 only SemanticSocial

#4 static + SemanticPersonal

#5 static + SemanticSocial



Accuracy: *Precision and Recall*

Precision(Pr): number of relevant selected items / number of selected items

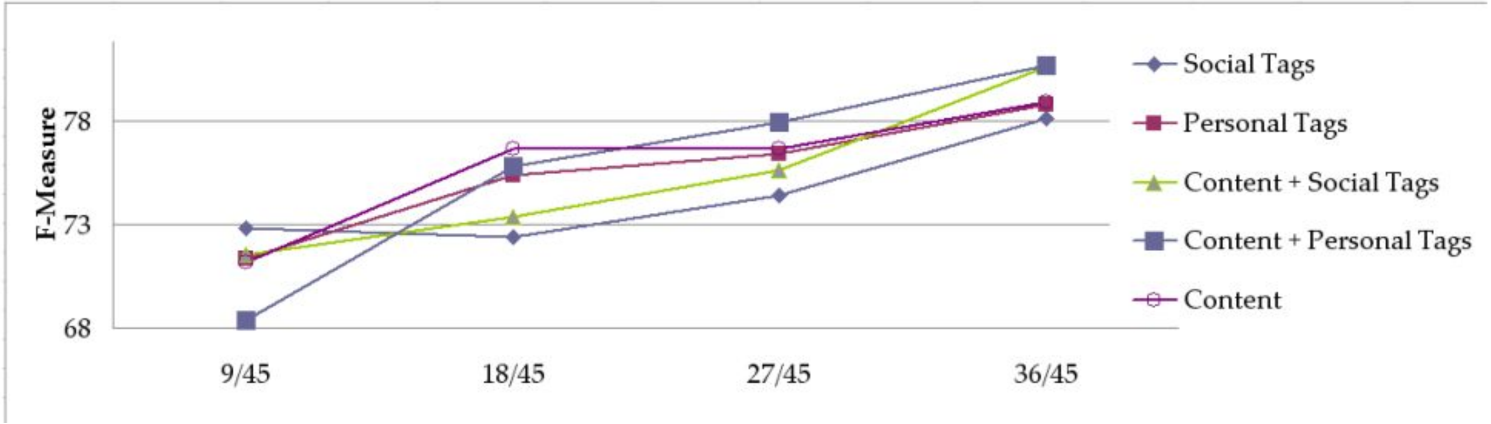
$$F_\beta = \frac{(1+\beta^2) \cdot Pr \cdot Re}{\beta^2 \cdot Pr + Re}$$

Recall(Re) : number of relevant selected items / total number of relevant items available

Results

Table 2: Results of the K-fold Cross Validation

Type of content	Pr	Re	$F_{\beta=0.5}$
Static Content	75.86	94.27	78.94
SemanticPersonalTags(U,I)	75.96	92.65	78.80
SemanticSocialTags(I)	75.59	90.50	78.17
Static Content+SemanticPersonalTags(U,I)	**78.04**	93.60	**80.72**
Static Content+SemanticSocialTags(I)	**78.01**	93.19	**80.64**



Conclusion

Main contribution:

Multivariate Poisson model for naive Bayes text classification adapted to infer user profiles

- In the end, using tags along with static information is better than recommending through just keywords or static information itself!
- perform an analysis of what tags are used to build the folksonomies and how they affect the user profile generation
- More diverse users

Thank you!