Ontology-Based News Recommendation

Wouter IJntema    Frank Goossen
Flavius Frasincar* Frederik Hogenboom

Erasmus University Rotterdam, the Netherlands

*frasincar@ese.eur.nl
Outline

Introduction

Hermes: News Personalization Service

Athena: News Recommendation Service

Athena Implementation

Evaluation

Conclusions and Future Work

Example
Introduction
Motivation

Problem

- Stock prices are sensitive to news
- News overload (different sources, different topics)
- Difficult to find the news of interest
- ... need for an intelligent solution to support news-based decision processes

Partial solution

- RSS feeds
- Broad categories (business, cars, entertainment, etc.)
Introduction

Motivation

Solutions

- News querying systems (intrusive)
- News recommender systems (non-intrusive)

Recommender systems:

- Content-based (Traditional)
- Collaborative filtering (Users-based)
- Semantics-based (Our focus here)
- Hybrid
Introduction
Related Work

- **Content-based**
  - Based on TF-IDF for representing articles and the user profile
  - Cosine similarity between new article and the user profile
  - Performance of cosine similarity decreases as the length of the article increases
  - Tools: YourNews, News Dude

- **Semantics-based**
  - Based on is-a relationships
  - Semantic relatedness as a similarity measure
    - Uses concepts instead of terms for the vector representation (improves precision)
    - Considers concepts related to the ones appearing in news items (improves recall)
  - Tools: PersoNews, (Getahun et al., 2009)
Hermes: News Personalization Service
Framework

- **Input:**
  - News items from RSS feeds
  - Domain ontology linked to a semantic lexicon (e.g., WordNet)
  - User query

- **Output:**
  - News items as answers to the user query

- **Four phases:**
  1. News Classification
     - Relate news items to ontology concepts
  2. Knowledge Base Updating
     - Update the knowledge base with news information
  3. News Querying
     - Allow the user to express his concepts of interest and the temporal constraints
  4. Results Presentation
     - Present the news items that match users query
Hermes: News Personalization Service
Architecture
Athena: News Recommendation Service
Framework

Input:
- News items from RSS feeds
- Domain ontology linked to a semantic lexicon (e.g., WordNet)
- User items of interest

Output:
- List of other news items of interest (possibly ranked)

Five similarity measures (alternatives):
- Concept Equivalence
- Binary Cosine
- Jaccard
- Semantic Relatedness (adaptation of (Getahun et al., 2009))
- Ranked Semantic Relatedness (our contribution)
Athena: News Recommendation Service
Preliminary Definitions

Ontology

\[ C = \{ c_1, c_2, c_3, \cdots, c_n \} . \] (1)

User Profile

\[ U = \{ c_{u1}, c_{u2}, c_{u3}, \cdots, c_{up} \} , \text{where } c_{ui} \in C . \] (2)

News Article

\[ A = \{ c_{a1}, c_{a2}, c_{a3}, \cdots, c_{aq} \} , \text{where } c_{aj} \in C . \] (3)
Athena: News Recommendation Service
Similarity Measures

Concept Equivalence

\[
\text{Similarity}(U, A) = \begin{cases} 
1 & \text{if } |U \cap A| > 0 \\
0 & \text{otherwise} 
\end{cases}.
\] (4)

- Concept Equivalence does not consider the number of user profile concepts found in a news article.

Binary Cosine

\[
B(U, A) = \frac{|U \cap A|}{|U| \times |A|}.
\] (5)
Jaccard

\[ J(U, A) = \frac{|U \cap A|}{|U \cup A|}. \] (6)

- Binary Cosine and Jaccard do not consider the number of occurrences of a concept in an article.
- Binary Cosine and Jaccard do not consider the concepts related to the ones found in an article.
Athena: News Recommendation Service

Similarity Measures

Semantic Relatedness

Semantic Neighbourhood

$$N(c_i) = \{c_1^i, c_2^i, \cdots, c_n^i\} \quad (7)$$

Vector Representation for 2 News Articles

$$V_l = (w_1^l, w_2^l, \cdots, w_p^l), \quad (8)$$

where

- $l \in \{i, j\}$, the two news articles $t_i$ and $t_j$
- $w_i$ represents the weight of $c_i$ (number of occurrences of $c_i$)
- $p = |CS_i \cup CS_j|$ is the number of distinct concepts in $CS_i$ and $CS_j$
Athena: News Recommendation Service

Similarity Measures

Semantic Relatedness

Vector Representation for 2 News Articles

\[ w_i = \begin{cases} 
1 & \text{if } \text{freq}(c_i \text{ in } CS_j) > 0 \\
\max_j(\text{ES}(c_i, c_j)) & \text{otherwise}
\end{cases} \quad (9) \]

where the enclosure similarity is defined as

\[ \text{ES}(c_i, c_j) = \frac{|N(c_i) \cap N(c_j)|}{|N(c_i)|}. \quad (10) \]

\[ \text{SemRel}(t_i, t_j) = \cos(V_i, V_j) = \frac{V_i \cdot V_j}{\|V_i\| \cdot \|V_j\|} \in [0, 1], \quad (11) \]
Athena: News Recommendation Service
Similarity Measures

Ranked Semantic Relatedness

Extended User Profile

- The set of related concepts to concept $c_i$ is

$$r(c_i) = \{c_i^1, c_i^2, \cdots, c_i^k\}.$$  \hfill (12)

- The set of related concepts to the concepts in the user profile is

$$R = \bigcup_{u_i \in U} r(u_i).$$  \hfill (13)

- The extended user profile is

$$U_R = U \cup R.$$  \hfill (14)
Athena: News Recommendation Service
Similarity Measures

Ranked Semantic Relatedness

Rank Matrix

<table>
<thead>
<tr>
<th></th>
<th>$e_1$</th>
<th>$e_2$</th>
<th>...</th>
<th>$e_q$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$u_1$</td>
<td>$r_{11}$</td>
<td>$r_{12}$</td>
<td>...</td>
<td>$r_{1q}$</td>
</tr>
<tr>
<td>$u_2$</td>
<td>$r_{21}$</td>
<td>$r_{22}$</td>
<td>...</td>
<td>$r_{2q}$</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>$u_m$</td>
<td>$r_{m1}$</td>
<td>$r_{m2}$</td>
<td>...</td>
<td>$r_{mq}$</td>
</tr>
</tbody>
</table>

where the ranks from the rank matrix are:

\[
 r_{i,j} = w_i \times \begin{cases} 
 +1.0 & \text{if } e_j = u_i \\
 +0.5 & \text{if } e_j \neq u_i, e_j \in r(u_i) \\
 -0.1 & \text{otherwise}
\end{cases} \quad (15)
\]
Athena: News Recommendation Service

Similarity Measures

Ranked Semantic Relatedness

Rank Matrix

- The weight $w_i$ is the number of articles the user has read about concept $u_i$.
- The elements of the rank vector $V_U$ for the extended profile concepts are:
  \[
  \text{Rank}(e_j) = \sum_{i=1}^{m} r_{ij} . \tag{16}
  \]
- The normalization of the rank vector $V_U$ is:
  \[
  V_U[v_i] = \frac{v_i - \min(v_u)}{\max(v_u) - \min(v_u)} . \tag{17}
  \]
Athena: News Recommendation Service

Similarity Measures

Ranked Semantic Relatedness

▶ A new article is a set of concepts

\[ A = \{a_1, a_2, \cdots, a_t\} \]  

(18)

▶ The rank vector of the article is

\[ V_A = (s_1, s_2, \cdots, s_t) \]  

(19)

where

\[ s_i = \begin{cases} 
\text{Rank}(e_i) & \text{if } e_i \in A \\
0 & \text{if } e_i \notin A 
\end{cases} \]  

(20)

\[ \text{RankedSemanticSimilarity}(V_A, V_U) = \frac{\sum_{v_a \in V_A} v_a}{\sum_{v_u \in V_U} v_u} \]  

(21)
Athena Implementation
Athena as HNP Plugin

- Hermes News Portal (HNP) is the implementation of Hermes
- Athena is a plugin for HNP
- Athena has three tabs:
  - Browser for all news items
  - Recommendations
  - Evaluation
- Implements all five recommenders
- Double clicking means the news item is added to the profile
Athena Implementation
HNP/Athena Implementation Tools

- Programming Language: Java
- Ontology Language: OWL
- Query Language: tSPARQL
- Semantic Web Framework: Jena
- Semantic Lexicon: WordNet
- Natural Language Processing: GATE
- Visualization: Prefuse
- Stemmer: Krovetz
Evaluation
Evaluation Setup

- 300 news items
- 5 users
- Each user has different interests
- All news items are marked as interesting/non-interesting by the users
- News items randomly split into two different sets:
  - Training set (60% of news items)
  - Validation set (40% of news items)
  - Similarity cut-off value: 0.5
## Evaluation Results

<table>
<thead>
<tr>
<th>Method</th>
<th>Accuracy</th>
<th>Precision</th>
</tr>
</thead>
<tbody>
<tr>
<td>TF-IDF</td>
<td>90%</td>
<td>90%</td>
</tr>
<tr>
<td>Concept Equivalence</td>
<td>44%</td>
<td>22%</td>
</tr>
<tr>
<td>Binary Cosine</td>
<td>47%</td>
<td>23%</td>
</tr>
<tr>
<td>Jaccard</td>
<td>93%</td>
<td>92%</td>
</tr>
<tr>
<td>Semantic Relatedness</td>
<td>57%</td>
<td>26%</td>
</tr>
<tr>
<td>Ranked</td>
<td>94%</td>
<td>93%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Method</th>
<th>Recall</th>
<th>Specificity</th>
</tr>
</thead>
<tbody>
<tr>
<td>TF-IDF</td>
<td>45%</td>
<td>99%</td>
</tr>
<tr>
<td>Concept Equivalence</td>
<td>98%</td>
<td>32%</td>
</tr>
<tr>
<td>Binary Cosine</td>
<td>95%</td>
<td>36%</td>
</tr>
<tr>
<td>Jaccard</td>
<td>58%</td>
<td>99%</td>
</tr>
<tr>
<td>Semantic Relatedness</td>
<td>92%</td>
<td>47%</td>
</tr>
<tr>
<td>Ranked</td>
<td>62%</td>
<td>99%</td>
</tr>
</tbody>
</table>
Evaluation
Evaluation Results

- Ranked Semantic Recommender scores better than TF-IDF for accuracy, precision, and recall, and the same for specificity

- Ranked Semantic Recommender scores best for accuracy and precision (closely followed by Jaccard)

- Ranked Semantic Recommender has a lower recall than Concept Equivalence, Binary Cosine, and Semantic Relatedness

- Concept Equivalence scores the best for recall
Conclusions and Future Work

Conclusions

- Athena: News Recommendation Service
- Athena implementation: HNP plugin
- Semantic recommenders are superior to traditional recommenders
- Ranked Semantic Recommender performs best for accuracy and precision
Conclusions and Future Work

Future Work

- Perform statistical significance tests

- Improve the recall of the Ranked Semantic Recommender by considering also the concepts related to the ones found in a new article

- Consider the indirect concepts in the semantic neighbourhood of a concept

- Refine the concept importance in an article: consider also the place appearance (title or/and body) in addition to number of occurrences
The user profile is:

\[ U = \{ \text{Yahoo!}, \text{Obama}, \text{China} \} \].

The weights \( W \) (number of articles) for the corresponding user profile concepts are:

\[ W = (4, 3, 2) \].

The sets of related concepts for each concept in the profile are as follows:

\[ r(\text{Yahoo!}) = \{ \text{Google, Apple} \}, \]
\[ r(\text{Obama}) = \{ \text{USA} \}, \]
\[ r(\text{China}) = \{ \text{USA} \}. \]
Ranked Semantic Recommender

Example

- The set of related concepts to the user profile concepts is:

\[
R = r(\text{Yahoo!}) \cup r(\text{Obama}) \cup r(\text{China})
\]

\[
= \{\text{Google, Apple, USA}\}.
\]

- The extended user profile is:

\[
U_R = \{\text{Yahoo!, Obama, China, Google, Apple, USA}\}.
\]

- The rank matrix is:

<table>
<thead>
<tr>
<th></th>
<th>Yahoo!</th>
<th>Obama</th>
<th>China</th>
<th>Google</th>
<th>Apple</th>
<th>USA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yahoo!</td>
<td>4</td>
<td>-0.4</td>
<td>-0.4</td>
<td>2</td>
<td>2</td>
<td>-0.4</td>
</tr>
<tr>
<td>Obama</td>
<td>-0.3</td>
<td>3</td>
<td>-0.3</td>
<td>-0.3</td>
<td>-0.3</td>
<td>1.5</td>
</tr>
<tr>
<td>China</td>
<td>-0.2</td>
<td>-0.2</td>
<td>2</td>
<td>-0.2</td>
<td>-0.2</td>
<td>1</td>
</tr>
<tr>
<td>Rank</td>
<td>3.5</td>
<td>2.4</td>
<td>1.3</td>
<td>1.5</td>
<td>1.5</td>
<td>2.1</td>
</tr>
</tbody>
</table>
Ranked Semantic Recommender
Example

- The normalized rank vector \( V_U \) is:

\[
V_U = (1, 0.5, 0, 0.091, 0.091, 0.364)
\]

- Two new news articles:

\[
A_1 = \{\text{Google, USA, Vitamins}\}
\]

\[
A_2 = \{\text{Yahoo!, USA}\}
\]

- The vector representations of these two articles:

\[
V_{A_1} = (0.091, 0.364, 0.0)
\]

\[
V_{A_2} = (1, 0.364)
\]
The ranked semantic similarities of these two news items to the extended user profile:

\[
\begin{align*}
\text{RankedSemSim}_{A_1} &= \frac{0.091 + 0.364}{1 + 0.5 + 0 + 0.091 + 0.091 + 0.364} \\
&= 0.222 \\
\text{RankedSemSim}_{A_2} &= \frac{1 + 0.364}{1 + 0.5 + 0 + 0.091 + 0.091 + 0.364} \\
&= 0.667.
\end{align*}
\]

- For a cut-off value of 0.5 only \( A_2 \) is recommended
- NB: Both \( A_1 \) and \( A_2 \) share only 1 concept with the user profile
Key Issues

▶ How to improve the recall for the Ranked Semantic Recommender?
▶ How to compute the importance of a concept in an article?