WebMate: A Personal Agent for Browsing and Searching

Liren Chen and Katia Sycara
Carnegie Mellon University

Introduction
- WebMate: an agent helping users to effectively browse and search the web
- Art
  - multiple TF-IDF vectors
  - Trigger Pair model for keyword refinement
  - Relevance feedback during the search
- using these techniques
  - provides effective browsing and searching
  - sends to users personal newspaper

Features
- Searching enhancement
  - Parallel search, searching keywords refinement, relevant feedback
- Browsing assistant
  - Learning, Recommending, Alias, Monitoring bookmarks, More like, Sending page to friends, Prefetching hyperlinks
- Offline browsing
  - downloading following pages, getting references and printing it out
- Filtering
  - Filtering HTTP header, cookie, block animation to speed up
- Checking html page: error finding, dead links,
- Dynamically setting up: search engines, dictionary
- Programming in java

WebMate architecture
- Learning user interests
  - a personalized newspaper
- Helping the user refine search
- system architecture

Learning profile to compile personal newspaper

Profile Representation and Learning Algorithm
- filtering task: judge whether relevant or not based on the user profile
- multiple user interests: single user profile, ask explicitly
- WebMate learns the categories automatically
- updates the profile incrementally and continuously
- other systems do not like this.
- Using TF-IDF with multiple vectors representation

\[ IDF(w) = \log \frac{DF(w)}{N} \]
\[ TF(w) = \frac{f(w)}{N} \]

1. Preprocess:
   - Parse HTML page, deleting the stop words, stemming the plural noun to single form, giving more weights to title word
2. Extract the TF-IDF vector for this document
3. If |V| < N (|V| is the number of vectors in the profile set V), then
4. Otherwise, calculate the cosine similarity between every two TF-IDF vectors including the vectors in the profile set V and the new document vector.

\[ \text{Sim}(V_i, V) = \frac{V_i \cdot V}{\|V_i\| \cdot \|V\|} \]

Multi TF-IDF vector learning
- \( N \): domains of interest (category) \( \{1, 2, \ldots \} \) - predefined number
- initial profile set \( V_0 \)
- \( M \): preset number of elements of a vector

for each positive example (“I like it”):
1. Preprocess
2. Parse HTML page, deleting the stop words, stemming the plural noun to single form, giving more weights to title word
3. Extract the TF-IDF vector for this document
4. If |V| < N (|V| is the number of vectors in the profile set V), then
5. Otherwise, calculate the cosine similarity between every two TF-IDF vectors including the vectors in the profile set V and the new document vector.

Assume the profile set V is

\[ \text{Sim}(V_i, V) = \frac{V_i \cdot V}{\|V_i\| \cdot \|V\|} \]

\( V \) is a feature vector representing the profile set V.

\( V_i \) is the feature vector for the positive example “I like it”.

\( V \) is the feature vector for the document.

\( \text{Sim}(V_i, V) \) is the cosine similarity between V_i and V.

\( \|V_i\| \) and \( \|V\| \) are the Euclidean norms of V_i and V, respectively.

\( \cdot \) denotes the dot product of the two vectors.
5. Combine the two vectors $V_1$ and $V_2$ with the greatest similarity

$$s = s_1 + s_2, \quad ||y|| = \sum_{i=1}^{n} ||y_i||$$

6. Sort the weights in the new vector $V_3$ in decreasing order and keep the highest $M$ elements

This algorithm is run whenever a user marks a document as “I like it”, thus the user profile is incrementally updated.

If the user does not provide any URLs that he would like to be the information sources, WebMate constructs a query using current profile.

Experiments

<table>
<thead>
<tr>
<th>Search refinement by keywords</th>
<th>expansion and relevance feedback</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trigger Pairs model to extract relevant words</td>
<td>single keywords are ambiguous</td>
</tr>
<tr>
<td>“stock” has more than 10 definition in the WordNet</td>
<td>manual expansion: user may not be able to provide the best refinement words</td>
</tr>
<tr>
<td>pruning: manual query expansion, semi-manual, automatic</td>
<td>“Best” = most frequently co-occur with the word in its intended meaning</td>
</tr>
<tr>
<td>“trigger pair” (S, T)</td>
<td>“trigger pair” (S, T)</td>
</tr>
<tr>
<td>If a word S is significantly correlated with another word T, then (S, T) is considered a “trigger pair”, with S being the trigger and T the triggered word</td>
<td></td>
</tr>
<tr>
<td></td>
<td>In the trigger Pairs Model (S, T) is different from (T, S), so the Trigger Pairs Model is different from the method of using co-occurrence of two words that is generally used in other keywords expansion</td>
</tr>
<tr>
<td>Mutual information (MI) : considers the words order</td>
<td>Mutual information (MI) : $M_1 = P(S,T) \log \frac{P(S,T)}{P(S)P(T)}$</td>
</tr>
</tbody>
</table>

Wireless News Corpus

set the maximum distance between S and T : 500

Trigger Pairs: sorted in decreasing order

- car {motor, auto, model, maker, vehicle, ford, buick, hond, inventory, assembly, chevrolet, sale, nissan, incentif, pontiac, planet, toyota, dealer, chrysler}

- music {simal, symphony, orchestra, composer, song, concert, tune, concerto, sound, musician, classical, album, violin, violinist, jazz, audience, conductor, play, audio, rock, cello, perform, dance}

Wall Street Journal Corpus

Trigger Pairs: domain specific

Keywords Expansion

Trigger Pair method can provide several candidate refinement keywords

1. $S_1 = \{s_1, s_2, \ldots, s_n\}$ is the trigger set and $X_1$ is the seed set of the user profile.
2. $S_2 = \{s_1, s_2, \ldots, s_k\}$ is the trigger set and $X_2$ is the seed set of the user profile.
3. $S_3 = \{s_1, s_2, \ldots, s_l\}$ is the trigger set and $X_3$ is the seed set of the user profile.
4. $S_4 = \{s_1, s_2, \ldots, s_m\}$ is the trigger set and $X_4$ is the seed set of the user profile.
5. $S_5 = \{s_1, s_2, \ldots, s_n\}$ is the trigger set and $X_5$ is the seed set of the user profile.
6. $S_6 = \{s_1, s_2, \ldots, s_k\}$ is the trigger set and $X_6$ is the seed set of the user profile.
7. $S_7 = \{s_1, s_2, \ldots, s_l\}$ is the trigger set and $X_7$ is the seed set of the user profile.
8. $S_8 = \{s_1, s_2, \ldots, s_m\}$ is the trigger set and $X_8$ is the seed set of the user profile.
9. $S_9 = \{s_1, s_2, \ldots, s_n\}$ is the trigger set and $X_9$ is the seed set of the user profile.
10. $S_{10} = \{s_1, s_2, \ldots, s_k\}$ is the trigger set and $X_{10}$ is the seed set of the user profile.
11. $S_{11} = \{s_1, s_2, \ldots, s_l\}$ is the trigger set and $X_{11}$ is the seed set of the user profile.
12. $S_{12} = \{s_1, s_2, \ldots, s_m\}$ is the trigger set and $X_{12}$ is the seed set of the user profile.
13. $S_{13} = \{s_1, s_2, \ldots, s_n\}$ is the trigger set and $X_{13}$ is the seed set of the user profile.
14. $S_{14} = \{s_1, s_2, \ldots, s_k\}$ is the trigger set and $X_{14}$ is the seed set of the user profile.
15. $S_{15} = \{s_1, s_2, \ldots, s_l\}$ is the trigger set and $X_{15}$ is the seed set of the user profile.
16. $S_{16} = \{s_1, s_2, \ldots, s_m\}$ is the trigger set and $X_{16}$ is the seed set of the user profile.
17. $S_{17} = \{s_1, s_2, \ldots, s_n\}$ is the trigger set and $X_{17}$ is the seed set of the user profile.
18. $S_{18} = \{s_1, s_2, \ldots, s_k\}$ is the trigger set and $X_{18}$ is the seed set of the user profile.
19. $S_{19} = \{s_1, s_2, \ldots, s_l\}$ is the trigger set and $X_{19}$ is the seed set of the user profile.
20. $S_{20} = \{s_1, s_2, \ldots, s_m\}$ is the trigger set and $X_{20}$ is the seed set of the user profile.

Broadcast News Corpus

- set the maximum distance between S and T : 500
- Trigger Pairs: sorted in decreasing order
- car {motor, auto, model, maker, vehicle, ford, buick, hond, inventory, assembly, chevrolet, sale, nissan, incentif, pontiac, planet, toyota, dealer, chrysler}

- music {simal, symphony, orchestra, composer, song, concert, tune, concerto, sound, musician, classical, album, violin, violinist, jazz, audience, conductor, play, audio, rock, cello, perform, dance}

- Trigger Pair method can provide several candidate refinement keywords

- Wall Street Journal Corpus
- Trigger Pairs: domain specific

- Trigger Pair method can provide several candidate refinement keywords

- if the user does not provide any URLs that he would like to be the information sources, WebMate constructs a query using current profile.

- Experiments

- Search refinement by keywords expansion and relevance feedback

- Trigger Pairs model to extract relevant words

- single keywords are ambiguous

- “stock” has more than 10 definition in the WordNet

- pruning: manual query expansion, semi-manual, automatic

- manual expansion: user may not be able to provide the best refinement words

- “Best” = most frequently co-occur with the word in its intended meaning

- “trigger pair” (S, T)

- If a word S is significantly correlated with another word T, then (S, T) is considered a “trigger pair”, with S being the trigger and T the triggered word

- In the trigger Pairs Model (S, T) is different from (T, S), so the Trigger Pairs Model is different from the method of using co-occurrence of two words that is generally used in other keywords expansion

- Mutual information (MI) : considers the words order

- Mutual information (MI) : $M_1 = P(S,T) \log \frac{P(S,T)}{P(S)P(T)}$
K1 = "charge" and S1 = {federal, investigation, attorney, plead, indict, ...} 
K2 = "fee" and S2 = {pay, dollar, million, bank, service, tax, raise, ...} 
K = (K1, K2) = "charge, fee" and S = S1 ∪ S2 = {million, pay, dollar, tax, service, federal, client, ...} 
  
- so triggers, such as million, pay, dollar, tax, service, help confine and disambiguate the meaning of the word "charge"

### Examples: keyword "stock"

- Yahoo! Stock Phrase Finder
- Stock Chart Finder
- Stock Phrase Finder
- Yahoo! Stock Phrase Finder
- Yahoo! Stock Phrase Finder

For example, the function defines K1 = stock share and S1 = {stock share} 

- stock share

### For example

- Suppose a user gives text as a relevant feedback to the search keywords "intelligent agent"
- URL: "http://www.cs.cmu.edu/~softagents"
- Using our method:
  - (software structure reusable architecture technology) used to expand the search "intelligent agent"