<table>
<thead>
<tr>
<th><strong>Title</strong></th>
<th>From More-Like-This to Better-Than-This: Hotel Recommendations from User Generated Reviews</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Authors(s)</strong></td>
<td>Dong, Ruihai, Smyth, Barry</td>
</tr>
<tr>
<td><strong>Publication date</strong></td>
<td>2016-07-17</td>
</tr>
<tr>
<td><strong>Conference details</strong></td>
<td>UMAP ’16, Halifax, NS, Canada</td>
</tr>
<tr>
<td><strong>Publisher</strong></td>
<td>ACM</td>
</tr>
<tr>
<td><strong>Item record/more information</strong></td>
<td><a href="http://hdl.handle.net/10197/9026">http://hdl.handle.net/10197/9026</a></td>
</tr>
<tr>
<td><strong>Publisher's statement</strong></td>
<td>© ACM, 2016. This is the author's version of the work. It is posted here by permission of ACM for your personal use. Not for redistribution. The definitive version was published in Proceedings of the 2016 Conference on User Modeling Adaptation and Personalization. <a href="https://doi.org/10.1145/2930238.2930276">https://doi.org/10.1145/2930238.2930276</a></td>
</tr>
<tr>
<td><strong>Publisher's version (DOI)</strong></td>
<td>10.1145/2930238.2930276</td>
</tr>
</tbody>
</table>
From More-Like-This to Better-Than-This: Hotel Recommendations from User Generated Reviews

Ruihai Dong
Insight Centre for Data Analytics
University College Dublin
Belfield, Dublin 4, Ireland
ruihai.dong@insight-centre.org

Barry Smyth
Insight Centre for Data Analytics
University College Dublin
Belfield, Dublin 4, Ireland
barry.smyth@insight-centre.org

ABSTRACT
To help users discover relevant products and items recommender systems must learn about the likes and dislikes of users and the pros and cons of items. In this paper we present a novel approach to building rich feature-based user profiles and item descriptions by mining user-generated reviews. We show how this information can be integrated into recommender systems to deliver better recommendations and an improved user experience.

Keywords
Sentimental Product Recommendation; Crowdsourcing

1. INTRODUCTION
Recommender systems help to provide users with the right information at the right time. They do this by learning about a user’s interests and preferences over time and use this profile information to select and/or rank items for recommendation, preferring those that are similar to those the user has liked in the past. Traditional recommendation approaches, such as collaborative filtering and content-based techniques, rely on product ratings or meta-data. Recently however researchers have consider user-generated review content based on features described in [2]. Each feature, \( f_j \) is associated with an importance score and a sentiment score as per Equations 2 and 3. An item description is composed of these features and scores as per Equation 1.

\[
\text{item}(h_i) = \{(f_j, s(f_j, h_i), \text{im}(f_j, h_i)) : f_j \in \text{reviews}(h_i)\} \quad (1)
\]

The importance score of \( f_j \), \( \text{im}(f_j, h_i) \), is the relative number of times that \( f_j \) is mentioned in the reviews of hotel \( h_i \).

\[
\text{im}(f_j, h_i) = \frac{\text{count}(f_j, h_i)}{|\text{reviews}(h_i)|} \quad (2)
\]

The sentiment score of \( f_j \), \( s(f_j, h_i) \), is the degree to which \( f_j \) is mentioned positively or negatively in \( \text{reviews}(h_i) \). Note, \( \text{pos}(f_j, h_i) \) and \( \text{neg}(f_j, h_i) \) denote the number of mentions of \( f_j \) labeled as positive or negative during the sentiment analysis phase.

\[
\text{s}(f_j, h_i) = \frac{\text{pos}(f_j, h_i) - \text{neg}(f_j, h_i)}{\text{pos}(f_j, h_i) + \text{neg}(f_j, h_i)} \quad (3)
\]

Similarly, we can generate a profile of a user \( u_q \) based on the reviews that they have written, by extracting features and importance information from these reviews as in Equation 4.

\[
\text{user}(u_q) = \{(f_j, \text{im}(f_j, u_q)) : f_j \in \text{reviews}(u_q)\} \quad (4)
\]

3. RECOMMENDATION RANKING
To begin with we implement a standard more-like-this approach in which we consider a query user \( u_q \) looking at some hotel \( h_0 \) and requesting similar items \( h_1, \ldots, h_c \). We use \( h_0 \) as the query and compare this to candidate items \( h_1, \ldots, h_c \), computing a similarity
score for each as the basis for ranking. Equation 5 demonstrates this for \( h_q \) and \( h_c \), using the importance scores of shared features as the feature values.

\[
Sim(h_q, h_c) = \frac{\sum_{f_i \in F(h_q) \cap F(h_c)} \text{im}(f_i, h_q) \times \text{im}(f_i, h_c)}{\sqrt{\sum_{f_i \in F(h_q)} \text{im}(f_i, h_q)^2} \times \sqrt{\sum_{f_i \in F(h_c)} \text{im}(f_i, h_c)^2}}
\]  

(5)

The above is a non-personalized similarity metric: the user’s profile has no bearing on the computation. We also implement a personalized version in which we use the importance weights from the query user \( u_q \) instead of the weights from \( h_q \) as in Equation 6.

\[
Sim_u(h_q, h_c) = \frac{\sum_{f_i \in F(h_q) \cap F(h_c)} \text{im}(f_i, u_q) \times \text{im}(f_i, h_c)}{\sqrt{\sum_{f_i \in F(h_q)} \text{im}(f_i, u_q)^2} \times \sqrt{\sum_{f_i \in F(h_c)} \text{im}(f_i, h_c)^2}}
\]  

(6)

Next we implement a better-than-this approach in which we include the sentiment score for candidate item. As mentioned earlier, sentiment information is unusual in a recommendation context but it’s availability makes it possible to consider not only how similar an item is to some query but also whether it enjoys a better sentiment value; we want to recommend items that are not similar to the query and have also been positively reviewed. We do this based on a feature-by-feature sentiment comparison as per Equation 7. We can say that \( f_i \) is better in a candidate item \( h_c \) than the query item \( h_q \) (\( \text{better}(f_i, h_q, h_c) > 0 \)) if \( f_i \) in \( h_c \) has a higher sentiment score than it does in \( h_q \). Then we can calculate the sentiment score, \( \text{Sent}(h_q, h_c) \) from the sum of these better scores for the features that are common to \( h_q \) and \( h_c \) as per Equation 8.

\[
\text{better}(f_i, h_q, h_c) = \text{sent}(f_i, h_q) - \text{sent}(f_i, h_c)
\]  

(7)

\[
\text{Sent}(h_q, h_c) = \sum_{f_i \in F(h_q) \cap F(h_c)} \text{better}(f_i, h_q, h_c) \times \text{im}(f_i, h_c) \bigg/ \big| F(h_q) \cap F(h_c) \big|\n\]  

(8)

Accordingly we can implement two scoring functions based on the above as per Equation 9: (1) a non-personalized version combining \( Sim_u \) and \( \text{Sent} \); and (2) a personalized version combining \( Sim_u \) and \( \text{Sent} \). We can adjust the relative influence of similarity and sentiment by using the parameter \( w \).

\[
\text{Score}(q, i) = (1 - w) \times Sim(q, i) + w \times \text{Sent}(q, i)
\]  

(9)

4. EVALUATION

The dataset used in this work is based on the TripAdvisor dataset [3] which covers 148,575 users, 1,088,585 reviews, and 1,701 hotels. For the purpose of this work we use a subset of 1,000 users with at least 5 hotel reviews for a total of 11,993 reviews for 10,162 hotels. For each of these hotels, we collected their top 100 reviews to produce a larger set of 867,644 hotel reviews.

For each of these users and hotels we apply opinion mining to generate feature-based descriptions. On average our test users have written 12 reviews resulting in profiles containing an average of 91 different features. Likewise the hotels are associated with an average of 89 reviews each, and in 189 features per review.

To evaluate our recommendation approaches we produce 888 test triples of the form \((u_q, h_q, h_t)\) corresponding to a query user \( u_q \), a query hotel from \( u_q \)’s profile, and a target hotel visited and rated as 5-star by \( u_q \). For each triple we use \( h_q \) (or \( h_p \) and \( u_q \) depending on approach) as an input and rank-order the other hotels in the same city as \( h_t \), using one of the two scoring variations, varying \( w \) to adjust the mix of similarity and sentiment. We compute how often \( h_t \) is within the top-20 of these ranked hotels.

The results presented in Figure 1 show that as we increase \( w \) (that is, increase the influence of sentiment over similarity) the hit-rate of both the personalized and non-personalized versions improves. For example, at \( w = 0 \) sentiment is not included in the recommendation scoring and we can see that the hit-rate falls between 0.26 and 0.30; meaning that the target hotel is found in the top-20 recommendations 26%-30% of the time. As we increase \( w \) up to about 0.5-0.6 then this hit-rate increases to between 0.35 and 0.38. Beyond this value of \( w \) the hit-rate begins to fall again. This tells us that the introduction of sentiment has a positive impact on recommendation quality, up to a point. Furthermore, we can clearly see how the personalized variation outperforms the non-personalized variations, by about 20%, particularly for values of \( w < 0.6 \).

5. CONCLUSIONS

In this short paper we have outlined an approach to recommendation based on user profiles and item descriptions that are mined from user-generated reviews. We have described how this approach allows us to mix similarity and sentiment during recommendation to demonstrate the value of both factors during recommendation. Furthermore, we have shown how this approach can also be used in a personalized recommendation setting.

6. ACKNOWLEDGEMENTS

This work is supported by Science Foundation Ireland under Grant Number SFI/12/RC/2289.

7. REFERENCES


