Content Ordering Based On Commuting Patterns

Travis Gingerich
Microsoft Corp.
Mountain View, CA
travisgi@microsoft.com

Omar Alonso
Microsoft Corp.
Mountain View, CA
omalonso@microsoft.com

ABSTRACT

Recommender systems take into account a wide range of information about both an individual user and other users' preferences in order to provide relevant content. However, one source of information that appears to be under-utilized is contextual information about the user's trajectory: where they are currently located, and where they are traveling to. We demonstrate a system that recommends the reading order of Twitter content based on the user's planned travel.

Categories and Subject Descriptors

H.4 [**Information Storage and Retrieval**]: Miscellaneous

1. INTRODUCTION

Recommender systems take into account a wide range of information about both an individual user and other users' preferences in order to provide relevant content; however, traditional content-based and collaborative filtering-based systems tend to rely on static user models [1]. As such there has been an increase in interest in context-aware recommender systems. One source of information that appears to be under-utilized is contextual information about the user's current travel. While there is previous work on recommendations from location-based social networks [2], little work has been done exploring trajectories.

With the current rising popularity of intelligent personal assistants such as Microsoft's Cortana and Google Now, mobile devices are increasingly aware of the context in which they are used. In particular, these assistants track users' daily commute patterns and also scheduled appointments, giving a wealth of contextual information that can be used to provide better content recommendation across a wide variety of content types. Our motivating scenario is as follows: a user commutes from home to work using public transportation and would like to consume relevant content as he/she travels to the destination. We are interested in exploring the *directionality* criterion, which assumes that users may be interested in entities that are on the future path rather than those that have been passed [3].

In this paper we present a recommender system that takes into account a user's current location and travel trajectory and provides a recommended reading order of their Twitter home stream (the stream containing all tweets and retweets from accounts the user follows). This ranking can be combined with a score from a rudimentary user model based on the user's tweets and favorite tweets to provide an alternative and personalized way to consume content apart from a strictly chronological ordering. Twitter is a source particular conducive to this location- and travel-aware ranking as many tweets contain hyperlocal content. While currently accessible through a browser, integration with an intelligent personal assistant is the intended use case for this ranking.

2. DEMO OVERVIEW

In this section we describe the main characteristics of the system. All content used by the system is obtained from the Twitter API[1].

2.1 Generating user model

The user model takes into account any tweets the user authored, retweeted, or marked as a favorite. Each tweet is stemmed using an implementation of the Porter stemmer, and the model consists of a vector of stemmed term frequencies. There are clearly many additional ways in which this model could be improved, but this serves as a proof of concept to show the feasibility of incorporating information about user travel with a content-based recommendation system.

2.2 Data processing

The application reads and stores tweets from the user's home stream over the most recent 5 hours. In order to take into account the user's travel, tweets must be mapped to particular locations. A relatively small number of tweets are geo-tagged or linked to a geo-tagged place by the tweet author. However, many account profiles have loosely-structured location information, and many tweets contain mentions of particular locations. We use an in-house named entity recognition system to identify locations, and map these and location information from the account profile to geographic coordinates using Bing Maps' geocoding API[2].

[1]https://dev.twitter.com/docs
[2]http://msdn.microsoft.com/en-us/library/ff701713.aspx

RecSys'14, October 6–10, 2014, Foster City, Silicon Valley, CA, USA.
ACM 978-1-4503-2668-1/14/10.
http://dx.doi.org/10.1145/2645710.2645715 .

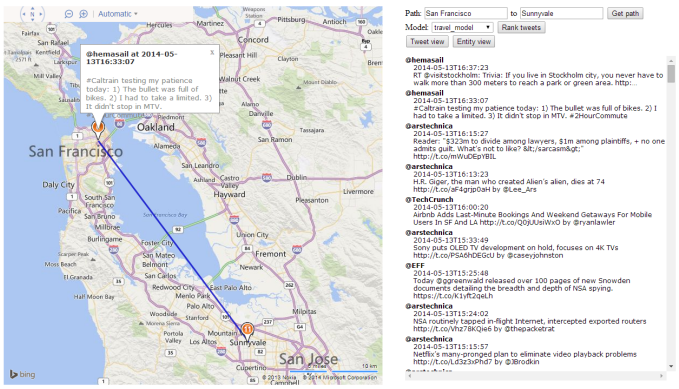

Figure 1: As the user travels from San Francisco to Sunnyvale, tweets near their current location in San Francisco are ranked higher than those closer to their destination.

2.3 Recommending a reading order

The demo uses two models. The first model (the "travel" model) asks the user to provide his/her current location and destination, and provides a reading order based on how each tweet falls along the user's approximate trajectory. For example, in Figure 1 the user is traveling between San Francisco and Sunnyvale. The trajectory is modeled as a straight line between the origin and destination. For each tweet, the ranking function identifies the closest point along this line to the tweet; tweets that fall at the user's current location along the line are given the highest score, and tweets closest to the destination are given a lower score. In this example, the user is currently in San Francisco, so tweets in the vicinity rank higher than those tweets that are located closer to the user's destination in Sunnyvale. This corresponds to the order in which the user will pass by the tweets' locations while traveling. The score of each tweet is penalized by exponential decay functions based on how far the tweet's location is from the closest point on the path, and the age of the tweet; this ensures that tweets that will fall in the user's vicinity rank higher than those that fall far away from the path, and that possibly stale tweets will be less likely to score highly. The equation is defined as: $score = \left(2 - \frac{d_1}{\text{path length}}\right) * e^{-\lambda_1 d_2 - \lambda_2 \alpha}$; d_2 is the distance from the closest point along the path to the tweet's location, d_1 is the distance from the start of path to that closest point, and α is the age of the tweet. λ_1 and λ_2 are configurable parameters that can be tuned to adjust how much tweets are penalized based on their age and distance from the path of travel.

The second model (the "blended" model) combines the score from the travel model with a score computed from the basic user model created for each user. The user model score is calculated using cosine similarity between the user model vector and a document vector derived from each tweet. As explained above, the user model consists of a vector in which the value for each token is the term frequency across tweets the user authored, retweeted, or marked as favorite: $w_{term,usermodel} = count(term)$. The tweet is modeled as a vector consisting of term frequencies, weighted by the inverse document frequency of the term (as seen across all tweets the system has consumed from the user's home stream): $w_{term,usermodel} = count(term) * \log(N/df_{terms})$ where N is the total number of tweets consumed from the stream and df_{terms} is the document frequency of the term across those tweets. The final score is the dot product between the two vectors:

$$score_{usermodel} = \sum_{term \in tweet} w_{term,usermodel} * w_{term,tweet}$$

The blended score is a linear combination of the user model and travel model scores:

$$score_{blended} = c_1 * score_{usermodel} + c_2 * score_{travel}$$

where c_1 and c_2 are parameters that adjust the relative importance of the user and travel models. The results are displayed using the same interface as the travel model. A video demonstrating the user interface is available online[3].

3. CONCLUSIONS AND FUTURE WORK

In addition to a web interface, the recommender system is also integrated with the yet-to-be-released Cortana intelligent personal assistant, providing the system a means to obtain information about commute patterns without requiring explicit user input. It is worth noting that this recommender system is not only limited to Twitter content; it can be applied to any content that has strong ties to a particular location, such as news or venues. The system we have presented here is a proof of concept showing how information about a user's travel plan can be taken into account to provide more relevant content ordering.

4. REFERENCES

[1] Gediminas Adomavicius, Bamshad Mobasher, Francesco Ricci, and Alexander Tuzhilin. Context-Aware Recommender Systems. *AI Magazine* 32(3), 2011, 67–80 .

[2] Max Sklar, Blake Shaw, and Andrew Hogue. Recommending Interesting Events in Real-time with Foursquare Check-ins. *Proc. of RecSys*, 2012, 311–312.

[3] Stefano De Sabbata and Tumasch Reichenbacher. Criteria for Geographic Relevance: An Experimental Study. *Int. Journal of Geographical Information Science*, 26(8), 2012, 1495–1520.

[3] http://aka.ms/bf3npo