SHOPAHOLIC: A Crowd-Sourced Spatio-Temporal Product-Deals Evaluation System (Demo Paper)

Kruthika Rathinavel
Department of Electrical and Computer Engineering,
Virginia Tech, USA
kruthika@vt.edu

Gaurav Dixit
Department of Computer Science,
Virginia Tech, USA
gdixit@vt.edu

Michael Matarazzo
Department of Computer Science,
Virginia Tech, USA
mfm11@vt.edu

Chang-Tien Lu
Department of Computer Science,
Virginia Tech, USA
ctlu@vt.edu

ABSTRACT

The emergence of internet advertising, email marketing and social networking has given rise to a new world of digital advertising used by stores and consumers alike. While retailers aim to promote all types of products, consumers also want to share this information via social media. This paper presents *Shopaholic*, a system that leverages social media to provide information on trending deals and store sales in any given location. It is intended to help shoppers identify great deals from the vast amounts of data scattered among social networks. Personalized search results, visualization of trends and sentiment analysis provided by *Shopaholic* allow the user to identify optimal deals. The application accounts for spatial and temporal data via a customized ranking algorithm and features integration with Twitter so that the user can share his or her actual experience using a deal. Ultimately, the system gives back to the shopping community by allowing users to share their experiences and evaluations of deals. A recommendation algorithm uniquely identifies the user's tastes, shopping history and current location to provide deal suggestions, thereby integrating temporal and spatial entities in recommendations.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*Information Filtering*

General Terms

Economics, Human Factors, Design, Experimentation

Keywords

crowd-sourcing, product-deals, consumer sentiment, social media

1. INTRODUCTION

In today's economy, we see many products ranging from groceries to electronics selling at large discounts. These discounts are generally represented in terms of percentages. However, the fidelity of these deals is often questionable. In most cases, shoppers assume that an advertised deal is the best deal available on the market. But since retail prices are usually marked up to begin with, no matter what the discount, the seller gains a profit while the shopper pays more for the item. Finding the optimal deal can be tricky and time-consuming. Many of these deals are hidden in the millions of tweets and messages scattered across social media where the consumers post about a great deal they found. Our application attempts provide a single system to bring together all the shopping deals posted on Twitter. By delivering crowd-sourced shopping deals to one location, it becomes easier for the shopper and boosts his or her confidence about a particular deal, given the approval from the crowd.

As we browse Google's Play Store or Apple's App Store, we see consumer-oriented applications like Shopular [2] and RetailMeNot [3]. These applications generally obtain their data via complex techniques that search the massive online databases of retailers, re-sellers, and small businesses. Shopular provides a collective space where current deals from the most popular stores are listed. RetailMeNot provides all available coupons for popular stores in one place. This information is obtained directly from the stores, and in most cases the applications require that the user have some type of store credit. In fact, some of these deals are not new; they simply reiterate information that shoppers are already aware of.

Shopaholic is developed as a tool to provide consumers with a single space for browsing available deals that are crowd-sourced from Twitter. User-generated information is largely scattered, and while several ideas are published and incorporated into live applications, none of them use crowd-sourced data to provide shopping deals and consumer sentiments to ease the shopping experience. Our major contributions are summarized as follows:

- **Sentiment Analysis and Spatio-Temporal Data Integration**: *Shopaholic* provides an interface that efficiently integrates textual and geographical representations of Twitter results (i.e. the deals) using a combination of spatial, temporal and textual processing techniques. *Shopaholic* also provides sentiment analysis on deals that the user typically looks for. Capturing other users' reactions to deals will help the user make an informed decision.
- **Visualization of Historical Data and Personalized Recommendations:** Visualizations of current and past trends allow the user to anticipate future deals and observe patterns of deals over a time span.
- **Personalized Recommendations:** Based on interesting profile data, location and past search history, *Shopaholic* recommends deals to the user after login and within search results.
- **Mobile Platform:** Crowd-sourcing data to Twitter, obtaining deal information and current trends via the ubiquitous mobile platform allows quick, convenient and on-the-go access to *Shopaholic's* key benefits; an improved consumer experience

Publication rights licensed to ACM. ACM acknowledges that this contribution was authored or co-authored by an employee, contractor or affiliate of the United States government. As such, the United States Government retains a nonexclusive, royalty-free right to publish or reproduce this article, or to allow others to do so, for Government purposes only.

IWGS'14, November 04-07 2014, Dallas/Fort Worth, TX, USA.
Copyright is held by the owner/author(s). Publication rights licensed to ACM.
ACM 978-1-4503-3139-5/14/11 $15.00
http://dx.doi.org/10.1145/2676552.2676558

and the opportunity for users to give back to the shopping community.

2. SYSTEM ARCHITECTURE

In this section, we describe the system architecture of *Shopaholic*. Figure 1 shows the three main components of the system: streaming of Twitter data, the application service and the mobile interface.

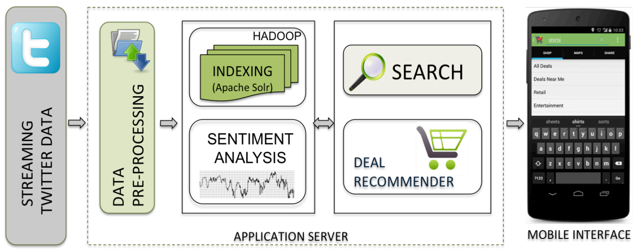


Figure 1. *Shopaholic* System Architecture

A. Streaming of Twitter Data.

This component performs real-time streaming and processing of Twitter data from the Washington D.C. metropolitan area. *Shopaholic* uses Twitter's streaming API to collect the tweets and filter them based on the user's location and key terms related to shopping. A parser then transforms the JSON data obtained into useful XML data and weeds out any unnecessary information collected from the Twitter server. The sentiment analyzer uses the coreNLP [5] package, processes every incoming tweet, performs sentiment analysis and appends the resulting sentiment information to the XML tree of each tweet.

B. Application Server.

This is the core server component of the application. It provides indexing, searching and ranking functionality. It also provides access to archived user information used by the recommender system. Because the recommender system accesses the user profile, previous search history information and indexed data, we split the underlying architecture to accommodate these separate tasks. The indexing, searching and ranking features are built using Apache Solr [6] as the underlying search engine. Apache Solr is an open source enterprise search platform with features including full-text search, hit highlighting, faceted search, dynamic clustering and a web administration interface. The recommender system uses the user profile information and current location to provide personalized recommendations to the user upon login and after searching for a deal.

The *Shopaholic* application is hosted on Apache Hadoop/HDFS 2.3 [7] making it highly scalable and robust. Since the project is currently in the initial phase, and we have only about 4.7 billion documents, we limit the server to one, thus running Hadoop in the pseudo-distributed mode (distributed, but with a single server). Apache Solr ensures reliability by allowing replication of the index. The system can run on a single index, scale to provide access to several thousands of users and still be reliable due to the replication mechanism.

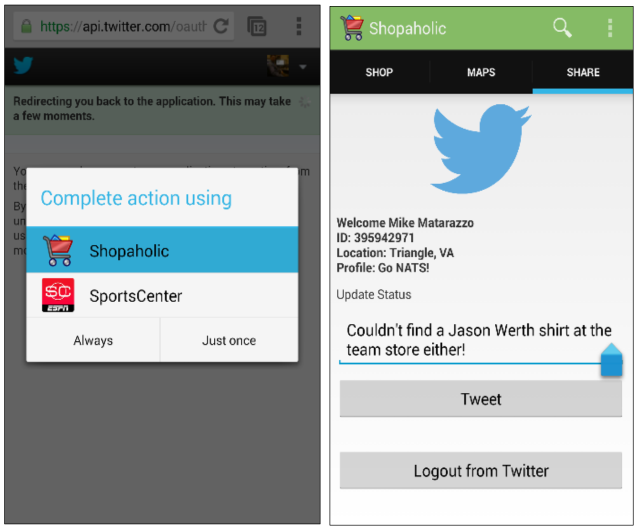
C. Mobile Interface.

This is the primary user interface of the application. It is developed as a native Android [10] application. The client-server interaction is accomplished using the HTTP protocol.

3. FEATURES

Shopaholic is a three-tabbed Android application that mobile users can use to search for current consumer deals. We develop a recommender system based entirely on crowd-sourced information from Twitter [1]. In addition to providing the user with current deals on a search item, the application also provides user experience information, i.e. user sentiments, for the results. Visualization of historical trends for a search item can help the user forecast deals and plan his or her shopping agenda.

3.1 Crowd-sourcing Real Time Data



Redirecting you back to the application. This may take a few moments.

Welcome Mike Matarazzo
ID: 395942971
Location: Triangle, VA
Profile: Go NATS!

Couldn't find a Jason Werth shirt at the team store either!

Figure 2. Logging In and Posting a deal to Twitter from *Shopaholic*

Shopaholic provides the users with real time deal information crowd-sourced from Twitter. Using Twitter's Streaming API [8], tweets are continuously streamed and indexed. The Twitter streamer uses a customized filter relevant to shopping deal information. This ensures that the user receives the most current deal information from Twitter. The latency for the latest data appearing on the application is roughly a few minutes, which is acceptable for shopping deals. Each tweet represents one or more possible deals, and is streamed, parsed, analyzed for user sentiments and then indexed on the Solr server. In the mobile interface, the SHARE tab allows users to login to Twitter via the *Shopaholic* application (Figure 2). Users can then 'share' a deal that he or she has discovered. This information will then be instantly posted to the user's Twitter account as his or her latest tweet. For tweets posted via *Shopaholic*, the hashtag *#shopaholicVT* is included. This facilitates further crowd-sourcing of deal information.

3.2 Sentiment Analysis and Integration of Textual, Spatial and Temporal Information

Shopaholic aims to provide real time deal information to users and therefore requires that, in addition to textual information, time and location information be processed and used for ranking. While indexing, the geo-location information is tagged appropriately before it is fed into the Solr engine. A comma separated string, providing latitude and longitude, is stored in the XML that is fed to Solr for indexing.

Upon launching the application, the user can choose from the categorical options available in the SHOP tab, or input a search keyword. Figure 3 displays some of the categorical options and shows how a user might use the top search. Figure 3 also shows a geographical view of the search results. A textual view of the results is shown in the form of tweets as shown in Figure 4.

A ranker is used to improve the quality of the results obtained from textual, spatial and temporal parameters. We provide a combination of three independent ranking methods and rank the final query results as a combination of these.

(1) The vector model's cosine similarity is used with modified weights for relevance between the textual query and the search results.

(2) The spatial ranking is based on Euclidean distance. We use the geospatial circular distance from a given point to provide the spatial filter for the search query. The given point here refers to the user's current location. We also use the Google Places API to improve geographical results by providing additional details about stores mentioned or related to the tweets. The query to the Google Places API is a HTTP 'get' request and uses spatial filters determined by the user's location. If this returns minimum results, we use the bounding box filter surrounding the calculated circle.

(3) We use the time filter for the search query to rank the result temporally.

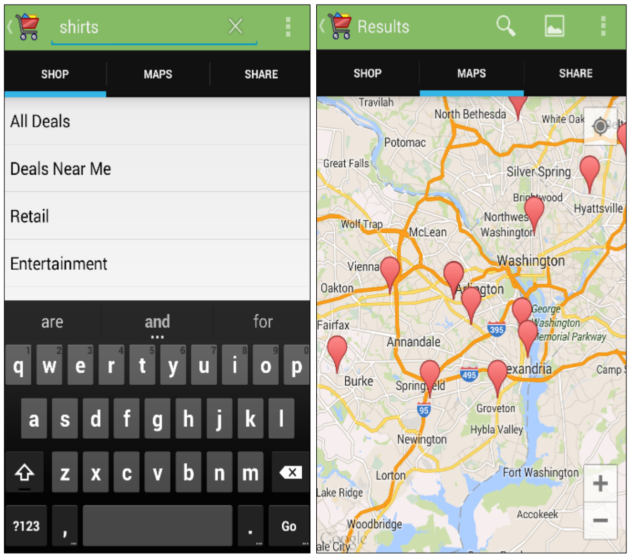


Figure 3. *Shopaholic* SHOP and MAPS Tabs

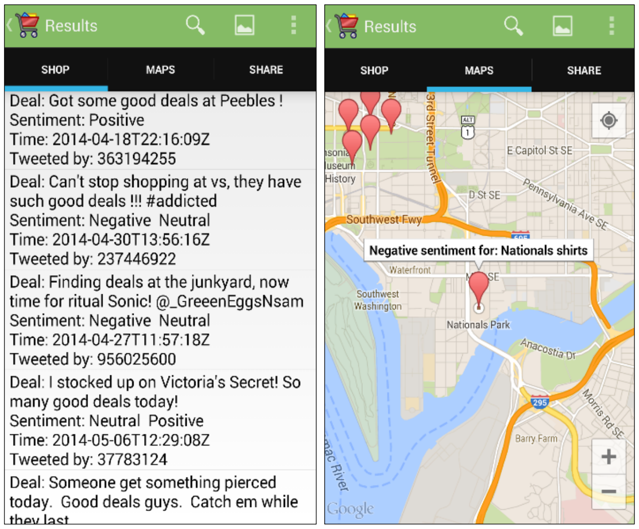
A weighted linear combination of these provides a score by which the results are ranked. The combined weighted ranking is used to rank the system.

$$R = w_t R_t + w_s R_s + w_{te} R_{te}$$

R_t, R_s and R_{te} are ranking parameters corresponding to the textual, spatial and temporal ranking, respectively, and w_t, w_s, w_{te} are weights associated with the textual, spatial and temporal ranking method, respectively. The weights w_t, w_s, w_{te} have the same range, normalized ranking scores are used, and the record is ranked higher when the normalized score is lower from the individual ranking methods. The latency of search results is attributed to the network speed of the user's mobile network.

Shopaholic analyses the sentiment of every tweet collected. Our sentiment analyzer is based on Stanford's coreNLP [5]. The terms in the tweet are first parsed to remove URLs, hashtags and Twitter handles. They are then tokenized, scored individually and the aggregate score is calculated for the entire tweet. In most cases, tweets are tokenized into phrases rather than individual terms. Pre-processing also includes sentence splitting, part of speech tagging, lemmatization and named entity tag annotation. The model is pre-trained to analyze tweets and categorize them as positive, negative or neutral by using the coreNLP API for each sentence in the tweet.

The final sentiment for every tweet is *positive*, *negative* or *neutral*. *Shopaholic* also provides an overall sentiment for the entire collection of results returned by a search. This gives the user a better idea of the overall quality of deals pertaining to a particular item. Figure 3 gives a list of deals along with the user sentiments; the user can also view the sentiments against a deal in the map view. Clicking a result in the text view takes the user to the corresponding tweet location in the map view. This gives the user a visual of where the deal corresponding to the tweet is located. A custom sentiment analyzer that considers shopping related aspects, including emoticons, is under development.



Deal: Got some good deals at Peebles !
Sentiment: Positive
Time: 2014-04-18T22:16:09Z
Tweeted by: 363194255

Deal: Can't stop shopping at vs, they have such good deals !!! #addicted
Sentiment: Negative Neutral
Time: 2014-04-30T13:56:16Z
Tweeted by: 237446922

Deal: Finding deals at the junkyard, now time for ritual Sonic! @_GreeenEggsNsam
Sentiment: Negative Neutral
Time: 2014-04-27T11:57:18Z
Tweeted by: 956025600

Deal: I stocked up on Victoria's Secret! So many good deals today!
Sentiment: Neutral Positive
Time: 2014-05-06T12:29:08Z
Tweeted by: 37783124

Deal: Someone get something pierced today. Good deals guys. Catch em while they last

Figure 4. Sentiment Assessments of Deals

3.3 Visualization of Historical Data

In the event that the textual list or map view of deals for a search query does not assist the user in deciding on any particular deal, *Shopaholic* provides a time-series plot of both trends for deals pertaining to the item or category searched as well as trends for the overall consumer sentiment of the result-set of deals. The time-series plots are adjustable and can cover a user specified date range. The plots can also display the popularity of deals for an item by showing the number of re-tweets.

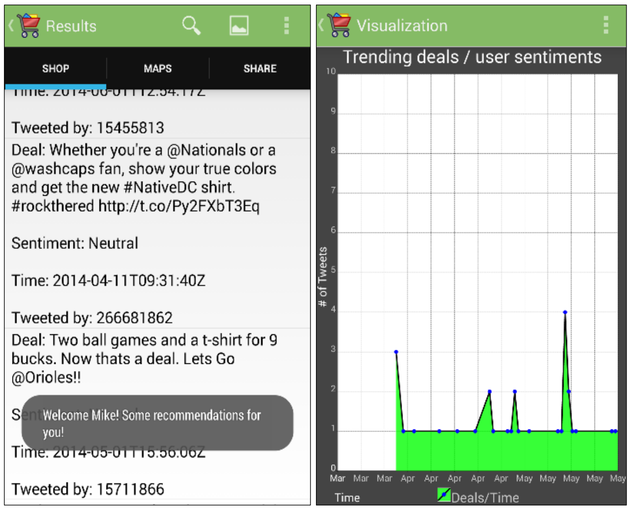
The visualizations are intended to consolidate important data for assisting the user with deciding on a deal. By showing current trends, historical data and statistics, the graphs would ideally allow users to predict upcoming "hot" periods for particular items that are not necessarily common sense, e.g. if a particular designer brand of shoes appears to have excellent deals at a certain time of year for no particular reason. As an example, Figure 5 shows how many deals are posted on average for the search term 'shirt' over the last few months. This gives the user an idea of how many deals might be available to search from.

Visualizations are created using the lightweight Plot API [9] in order to avoid the overhead of a more robust data analytics API for Android, of which there are not many. It may help the user to know not only where the deals are, but also what the hot venues are and how much time they have to take advantage. An Android-based Google Map, provided by Google Play Services, is used to provide a geographical layout of search results.

3.4 Personalized Recommendations

When a user logs in to *Shopaholic*, he or she sees a list of recommendations based on various parameters collected from previous visits (Figure 5). For a first time user, *Shopaholic* captures user profile information and tweet history to provide a basic list of recommended deals.

Apache Lucene, which powers Solr, is a multi-dimensional sparse matrix with very fast look up capabilities. We utilize this feature in combination with a clustering algorithm to define our recommendation system. The recommendation is also ranked temporally, giving the user the latest deals available, and can incorporate search history.


Results
SHOP
MAPS
SHARE
Visualization
Trending deals / user sentiments
10
9
8
7
6
5
4
3
2
1
0
# of Tweets
Mar Mar Mar Apr Apr Apr Apr Apr May May May May May
Time
Deals/Time

Time: 2014-06-01T12:54:17Z

Tweeted by: 15455813

Deal: Whether you're a @Nationals or a @washcaps fan, show your true colors and get the new #NativeDC shirt. #rockthered http://t.co/Py2FXbT3Eq

Sentiment: Neutral

Time: 2014-04-11T09:31:40Z

Tweeted by: 266681862

Deal: Two ball games and a t-shirt for 9 bucks. Now thats a deal. Lets Go @Orioles!!

Welcome Mike! Some recommendations for you!

Time: 2014-05-01T15:56:06Z

Tweeted by: 15711866

Figure 5. Recommendations After User Login and Trends of Deals / User Sentiments

When the user logs in, *Shopaholic* collects the profile information about the user and uses it to find deals relevant to the current location and previous tweets related to shopping (if any). When the user types in the first keyword to search, this keyword is stored in a local user profile for the mobile application. This keyword list grows every time the user types in a key word to search. A weight is assigned to each keyword based on the number of times it is searched. The list looks as follows:

Bike rack (15), Helmet (9), Hiking shoes (5), …

In combination with the user's Twitter profile and current location, this list is used to provide recommendations to the user. As the list grows, we generate clusters. *Shopaholic* uses a customized clustering algorithm derived from Carrot2 clustering algorithm [4]. We apply the clustering algorithm to the tweet, hash tags and location from where the tweet was posted.

The pre-processing stage is the same as that of document indexing. This includes using Porter Stemmer for stemming and a customized stop word list. Under feature extraction, we discover phrases and single terms that can form a cluster-label for the abstract concepts to be discovered. We identify frequently used terms and phrases [11] using a variant of suffix arrays extended with a longest common prefix array. We use a term frequency threshold to choose the terms and phrases whose frequency exceeds the threshold value.

For the next step, we build the term-document matrix, and then perform Singular Value Decomposition [12] based on the matrix to obtain U, Σ and V matrices. Using Σ matrix, we obtain the k abstract concepts. We obtain the U_k matrix by using first k columns of the U matrix, and perform *tf-idf* weighting to obtain phrase matrix P. To find best matching phrase, we perform $U_k \times P$ for every column in U_k matrix, and find the largest value of resulting vector. Then the similarities between all pairs of labels are calculated. We again use a similarity threshold to find groups of labels that are above this threshold. For every group, we find one label with the highest score, and this will be the cluster label. Cluster content is identified using the Vector Space Model, which we modify by matching the input snippets against every single cluster label instead of just matching with a single query, we match it against every single cluster label. Finally cluster scores are calculated as:

$$cluster\ score = label\ score * member\ count$$

4. SUMMARY

Shopaholic is developed to meet the needs of the user who is looking cautiously to pick the right deal without falling prey to discounter-marked up prices. Our unique combination of customized algorithms and integration of spatial, temporal and textual information provides users with real time information on useful shopping deals. In addition to search, the user is also provided with personalized recommendations of the latest deals, and can also share a deal he or she likes or has used and found to be worthy of using. From this, other *Shopaholic* or Twitter users can benefit. The recommender system and the crowd sourcing of information make *Shopaholic* a compelling system for end users. We envision *Shopaholic* to be the go-to platform for browsing shopping deals. By sharing more deals, consumers can spend less and shop more, making the entire shopping experience more enjoyable.

A video demonstration of *Shopaholic* can be viewed at the following YouTube URL: http://www.youtube.com/watch?v=qo5tvCAgyKM

5. REFERENCES

[1] *Twitter*. Available from: https://about.twitter.com

[2] *Shopular*. Available from: http://www.youtube.com/watch?v=jNGv0ehFiJE

[3] *RetailMeNot*. Available from: http://www.retailmenot.com

[4] Osiński, S. and Weiss, D. 2005. Carrot2: Design of a Flexible and Efficient Web Information Retrieval Framework. *Springer Berlin Heidelberg, Advances in Web Intelligence* (2005), 439-444.

[5] *CoreNLP*. Available from: http://nlp.stanford.edu/software/corenlp.shtml

[6] *Apache Solr*. Available from: https://lucene.apache.org/solr/features.html

[7] *Apache Hadoop/HDFS*. Available from: http://hadoop.apache.org/docs/r1.2.1/hdfs_design.html

[8] *Twitter Streaming API*. Available from: https://dev.twitter.com/docs/api/streaming

[9] *Android Plot*. Available from: http://androidplot.com

[10] *Android SDK*. Available from: http://developer.android.com/guide/index.html

[11] Dell Zhang and Yisheng Dong. Semantic, Hierarchical, Online Clustering of Web Search Results. Accepted by 3rd International Workshop on Web information and data management, Atlanta, Georgia, 2004, 69-78.

[12] *Computation of a singular value decomposition*. Available from: http://www.cs.utexas.edu/users/inderjit/public_papers/HLA_SVD.pdf