Mine & Dine

Combining Collaborative Filtering and Social Media to Make Better Restaurant Recommendations

Kunal K. Lala
klala@seas.upenn.edu
Univ. of Pennsylvania
Philadelphia, PA

Susan Davidson
susan@cis.upenn.edu
Univ. of Pennsylvania
Philadelphia, PA

ABSTRACT

Currently, to find a restaurant recommendation, a user may go online and visit one of many popular review-aggregation websites (Yelp and TripAdvisor are two popular examples). Rather than explicitly make recommendations, however, these sites present the user with an expansive list of local options and their average ratings- calculated from members' reviews. These sites lack personalization of restaurant ratings for individual users, and the sites' review pools are often small, noisy, and unrepresentative of the true quality of the restaurants they describe.

We describe a system that seeks to correct these shortcomings. Mine & Dine is a restaurant recommendation engine that mines web-based social media to make unique recommendations based on other restaurants the user has already visited. The user enters as input a variable number of restaurants he or she has already visited in the area, and then ranks each one on a real number scale from 0 to 5. The system parses through web logs (or "blogs" for short) and tags those that qualify as reviews of local restaurants. From these reviews, item-based collaborative filtering is employed to predict the user's ratings of other restaurants in the area and rank them. The highest-ranked restaurants are then recommended to the user.

To evaluate our recommendations, we calculate the mean absolute percentage error (MAPE) of a users' predicted restaurant ratings in test data. Comparing Mine & Dine's performance to recommendations based on average restaurant ratings (the baseline), we have achieved a significant 10% reduction in MAPE.

1. INTRODUCTION

1.1 Background

Today's Information Age has seen an explosion of activity in the realm of social media. With the greater interconnectivity that blogs and online review sites offer, an increasing number of people have begun publishing and sharing their opinions online. This pool of opinions has turned into an invaluable resource for both savvy consumers and retailers alike. The former looks to opinion-rich social networks for recommendations and guidance on shopping, travel, and purchase decisions. The latter seeks to mine these networks for new marketing opportunities.

According to Pang and Lee [11], two surveys of over 2000 American adults showed that "81% of Internet users (or 60% of Americans) have done online research on a product at least once; 20% (15% of all Americans) do so on a typical day; and among readers of online reviews of restaurants, hotels, and various services (e.g., travel agencies or doctors), between 73% and 87% report that reviews had a significant influence on their purchase." Yet, despite these encouraging statistics, "58% [of American internet users] also report that online information was missing, impossible to find, confusing, and/or overwhelming" [5].

Online review-aggregation sites like *Yelp*, *TripAdvisor*, and *UrbanSpoon* can take credit for enabling savvy consumers by offering them a community and forum to share their opinions. These websites became invaluable repositories of data; they rapidly built large user bases and amassed shared content. Unfortunately, these companies have not done much to utilize this content beyond standardizing its reporting and improving its accessibility. The trend in entrepreneurial effort and research has been in engaging the vast untapped data that is being generated through social media sites.

The most easily identifiable building block of the social media boom is undoubtedly the blog. Blogs are personalized websites through which users publish their thoughts, opinions, and commentaries on a diverse spread of topics. Many function as online journals or diaries. They vary in structure and length (new micro-blogs allow for short bursts of communication- Twitter is a well-known example). Given that the "blogosphere" (the entire set of the internet's blogs) spans millions of users' sites in size, there is certainly a sizable dataset waiting to be mined. One of the newest ways researchers have found to utilize this data is to mine it for opinions- to take the "pulse of the web."

Sentiment analysis (also referred to as opinion mining) involves examining text to identify the author's general underlying viewpoint or opinion with regard to a particular entity. The real goal and challenge of sentiment analysis is to correctly classify a document's viewpoint (or polarity) as positive, negative or somewhere in-between [10]. Numerous companies are looking into sentiment analysis techniques to get a better idea of their brand's perception in the public. Others have monitored the web's opinions regarding certain stocks and used them to make big profits on the margins. Regardless of its specific application, sentiment analysis has garnered much attention as the key to making sense of all the data that has filled the web.

While sentiment analysis is used to extract reviews from blogs, collaborative filtering techniques are used to extract personalized recommendations from these reviews. More specifically, collaborative filtering (or "CF") seeks to determine items that a user will like based off of his ratings of other items and off opinions of other like-minded users. Traditional user-based CF algorithms work by building a matrix that maps item preferences to users. A user's predicted ratings of items he hasn't yet experienced can be calculated by finding people in the matrix who have traditionally liked the same kinds of items as the user and using their ratings as a guide. A variation of user-based CF that yields better average performance is the item-based CF method [12] . Rather than computing similar "neighbors" to a user every time he or she request a recommendation, this method calculates similarities between the items themselves. This involves a fairly expensive calculation, but because relationships between the items do not change often, similarities only need to be recomputed occasionally.

1.2 Shortcomings of Current Recommendation Sites

There are many ways in which current review-aggregation websites are imperfect. Here, we discuss their key shortcomings.

1.2.1 Choice Paralysis

Presentation of recommendations in status-quo systems is not conducive to quick selection. Recommendations provided are difficult to disambiguate; given a set of "excellent", 5-star restaurants, there is no certain criterion that a user can use to choose one restaurant over another. For example, if a user wants Thai food in Philadelphia, *Yelp* will provide that user with a list of 5-star Thai restaurants and let the user sort out the specifics. These results are numerically ranked, but when they all share 5-star rankings, extra points on the margin are not enough to allow a user to confidently pick one restaurant over another.

1.2.2 Sparsity and Noise

One of the inherent limitations of membership-based review aggregation websites like *Yelp* is that restaurants' reviews are limited to those submitted by members. As a result, there are restaurants that receive their mean rating from a very small set of reviews. When the review set is small, unhelpful reviews (those that don't reflect the true quality of the venue) skew the overall rating of the restaurant.

1.2.3 Bimodal Ratings Distributions

According to research done by Hu, Pavlou, and Zhang [6], current review engines are skewed towards extreme values with a bias towards positive reviews. Distributions of these data often don't appear as normal bell-curves. Rather, a majority of ratings distributions appear bimodal. The intuitive explanation for these distributions is that users are inspired to leave reviews only when their experience was wonderful or terrible enough to motivate them thoroughly.

1.3 Mine & Dine

Mine & Dine seeks to correct the aforementioned problems by expanding the review set from the member-base of any review aggregation site to the blogosphere. To do so, we utilize restaurant review data pulled from *CitySearch*, a site that indexes and normalizes reviews across the web for venues in different U.S. locales. Because reviews for each restaurant are pulled from numerous official and unofficial web sources, processing these reviews is effectively a coarser form of processing user opinions on the blog-level of granularity. By expanding the set of reviews, we minimize the effects of noise. Also, assuming bloggers are motivate by the desire to recount their day more than anything else, their opinions about restaurants hopefully will not be affected by the "brag or moan" phenomenon that causes bimodal ratings distributions. Finally, through an item-based collaborative filtering process, Mine & Dine eliminates choice paralysis on the user's part by generating one best venue recommendation. Evaluating our proposed system, we see a significant improvement in mean absolute percentage error (MAPE) over the baseline.

2. RELATED WORK

2.1 Research on Opinion-Based Decisions and Purchases

Some of the first interesting research that is related to present-day semantic analysis was conducted by Nelson in 1970 [9]. He created a dichotomy of purchase types; one category of purchases were heavily dependent upon others' opinions and the other category could be purchased independent of others' recommendations. He showed that the value of others' opinions is proven to be very high when dealing with goods that are bought based on the "information process experience" rather than those that are bought through the "search" information process [9]. These "search goods" can be evaluated by the user alone prior to purchase. "Experience goods" are not evaluable prior to purchase [9]. Senecal and Natel [13] make the argument that, since the quality of experience goods cannot be determined prior to purchase, consumers should rely more heavily on the recommendations of others for these kinds of goods rather than for goods with "search qualities."

From this information, we can extrapolate that consumers probably will not rely on recommendation/rating systems (let alone others' opinions) for items that they themselves can afford to adequately examine before purchase. Senecal and Natel [13] provide an example of how consumers assessing the purchase of 35-mm camera will use their own decision-making processes rather than relying on others for help. For deciding which restaurant to visit, however, a consumer relies on the opinions of others for deciding which of the many offerings to choose from; there is no real way to tell if a restaurant's food will suit your palette just by examining the menu endlessly. This can help explain the surge in popularity of online review sites like *Yelp* . People want others' opinions when settling on an experience-based good like a restaurant or entertainment venue.

2.2 History of Sentiment Analysis Research

In 1992, Marti Hearst [4] described a "Text-Based Intelligence System" that would allow a text corpus to be classified by its directionality (a classification criterion based on whether an agent in a text is in favor of, neutral, or opposed to a particular event). Hearst seized upon the idea that documents could be classified without resource-intensive and time-consuming full-sentiment analysis. Rather, she proposed a method of extracting only the semantic data necessary to classify a document's directionality [4]

In 1994, Wiebe [14] presented an algorithm used for tracking point of view in narratives based on regularities found in the ways authors presented point of view, and in 1995, Wiebe and Bruce [15] expanded on this work by segmenting text into blocks of subjective sentences and linking the point of view expressed by this block to a particular agent.

Pang and Lee [11] point to 2001 as the time when research on sentiment analysis and opinion mining really took off. They attribute this surge of interest to the popularity of machine learning algorithms applied to natural linguistic processing and to the increased availability of large data sets on which to train their classifiers.

A year later, Morinaga and Yamanishi [8] proposed a product reputation mining system that utilized a binary (positive or negative) dictionary to type words associated with a product on mined web pages. The novel aspect of their work was labeling these opinions with an opinion-likeliness calculation of 1-5, which signified just how likely it was that the opinion actually was an opinion. Then, characteristic words were extracted from these opinions to help differentiate them and construct a reputation for a given product. The limitations of this research involved dependence on the user to provide analysis conditions as well as significant reliance on pre-tagged data (for determining the positivity or negativity of web pages, for extracting characteristics, etc.).

Dave, Lawrence, and Pennock [2] produced a classifier that utilizes feature extraction and scoring to tag website opinions about a particular product as "poor, mixed, or good". They trained their classifier on user reviews from C|net and Amazon. While their classifier performed decently, they ran into problems of rating inconsistency and a sparse data set. As their research was conducted in 2003, they didn't have access to the sizable data sets that have just recently become available through donations and concerted data mining efforts.

2.3 Recent Work

Recent work in the field of sentiment analysis seems to have focused mainly on bettering the training of classifiers and finding smarter ways to extract and tag the non-neutral, opinionated portions of documents. There is also interesting new work in the field of aspect-based sentiment summarization, in which "textual evidence" is extracted from user reviews in support of their rating of a venue.

Pang and Lee [10] explored the issue of sentiment analysis using polarity classifiers on just the subjective portions of documents. The key to extracting these portions is through a novel technique that finds the minimum cuts in graphs. Their naive Bayes classifier actually performed better on the extracted data then on the original documents, suggesting that their extraction technique removed unwanted noise in addition to trimming the documents.

Wilson, Wiebe, and Hoffman [16] utilized a phase-level approach to determine the sentiment of a document by first checking whether expressions were subjective or objective, and then categorizing the subjective expressions by their contextual polarity. Using this method, they were able to achieve good results above the baseline.

Ryan McDonald and Ivan Titov released a research paper titled "A Joint Model of Text and Aspect Ratings for Sentiment Summarization" in which they propose a general model of extracting textual evidence to supporting individual ratings of restaurant aspects- like food, decor, service, and value. Their model relies upon numerical ratings and a textual review of these aspects, but the text does not have to be directly associated with an aspect. The strength of this joint model is that it can pull the relevant text associated with an aspect from a general review as long as individual aspect ratings are provided [7].

A hybrid model of aspect extraction and its corresponding architecture for parsing reviews and producing final aspect summaries is presented in [1]. In this paper, researchers attempted to generalize the idea of *aspects* into *services*, so that their textual summaries would be relevant even for entities like "hair salons, schools, museums...". The architecture they propose is very interesting and applicable to the kind of summaries we hope to provide in our own recommendation system. For every review of a local service, they first identify all the sentences they tag as expressing sentiment. Then, they identify relevant aspects for a service that are mentioned in these tagged text fragments. Finally, they aggregate the sentiments over a particular aspect dependent upon the sentiments expressed in those fragments. They achieve a high level of accuracy in linking textual evidence to a review's aspects, and they credit their hybrid approach to raw labeling and a maximum entropy model.

Work by Hu, Pavlou, and Zhang [6] is especially interesting because it calls into question the ability of online reviews to adequately reflect the quality of any product. They describe how most online review site contributors are motivated to write reviews in order to "brag" or "moan"; this is an interpretation of how online reviews seem to fall either in the positive or negative extremes of the rating spectrum. They discovered that the rating distributions for different products were not bell-shaped curves like they expected. Rather, 53% of the online product rating distributions they examined were bimodal, with results skewing towards the positive [6]. The significance of this research is that average ratings do not reflect a convergence of ratings so much as they do a balance between two extreme camps of reviewers. They hypothesize that unless all consumers are forced to review a product or there is a "symmetric impact of bragging and moaning for consumers leaving review", the average rating for a product will not be indicative of its quality.

There are websites out there that act as toolsets for conducting sentiment analysis. SentiWordNet is one such free web tool that trains ternary classifiers using semi-supervised learning to tag words in *synsets* as positive, negative, or objective [3].

Surveying websites that provide itinerary and venue recommendations, we could find none at the time of writing this paper that match the kind of service we plan to offer. TripIt comes close by offering itinerary planning based off of a user's travel plans, but the recommendations are not pulled from the blogosphere. *Yelp* and *UrbanSpoon* are two examples of review-aggregation sites that have been built from users' reviews, but they, too, stop short of tapping a wider audience for reviews.

3. SYSTEM MODEL

3.1 Overview

Mine & Dine takes as textual input the names of restaurants the user has already visited along with real number rankings from 0-5 for each restaurant. It accesses a restaurant similarity matrix to find similarity scores between the restaurants entered and all the other restaurants in the area.

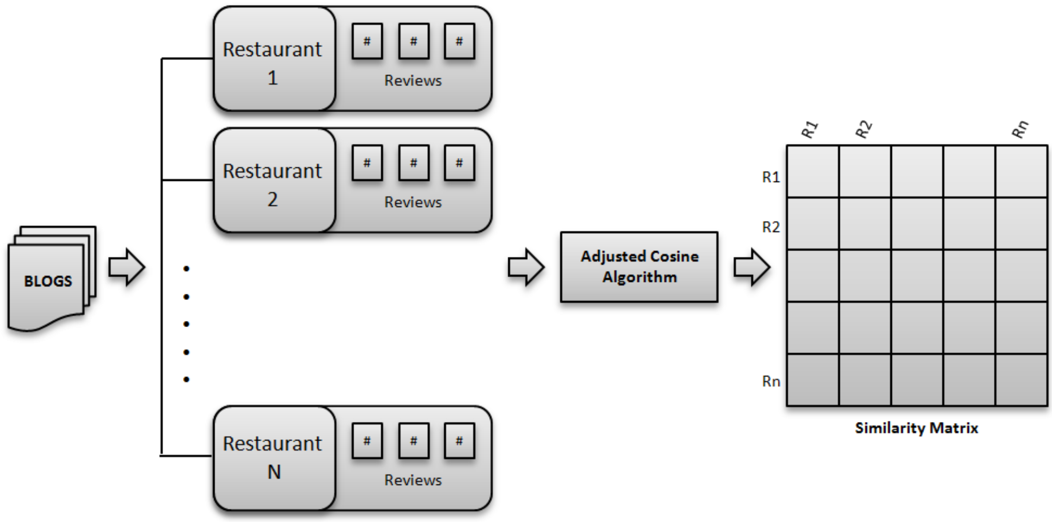

Figure 1: The item-based collaborative filtering process

Then, using these similarity scores and the user's ratings of the visited restaurants, the system generates the user's predicted ratings for all other restaurants. These predicted scores are ranked from greatest to least, and the the top-ranked restaurant is recommended to the user. The bulk of computation is spent building the restaurant similarity matrix; the blog data has to be parsed and similarity scores have to be calculated for every pair of restaurants in the user's locale. As explained earlier, however, the key benefit of item-based collaborative filtering is that calculating item-item similarities- though computationally expensive- need only be done once in a while. Paying this computation up-front allows for better average performance during the actual recommendation generation process. We break up recommendation generation into calculation of restaurant similarities and then making recommendations based off of those similarities.

3.2 Restaurant Similarity Calculation

To build the similarity matrix, we follow the steps in Figure 1 and parse through the review dataset and map restaurants to the users that reviewed them and map users to the restaurants they reviewed. These mappings are necessary because the similarity rating between a restaurant i and a restaurant j depends on the subset of reviewers who have reviewed both restaurants. Though there are a number of ways to calculate similarity, we use the adjusted cosine method which treats two restaurants as vectors and their similarity as the angle between those vectors. What differentiates the adjusted cosine similarity calculation from the more conventional cosine calculation is the adjustment for the difference in ratings scales of different users. For example, one user's definition of a 5-star restaurant is another user's 3-star restaurant. Sarwar *et. al* give the following formal equation for calculating adjusted cosine similarity:

$$sim(i,j) = \frac{\sum_{u \in U}(R_{u,i} - \bar{R_u})(R_{u,j} - \bar{R_u})}{\sqrt{\sum_{u \in U}(R_{u,i} - \bar{R_u})^2}\sqrt{\sum_{u \in U}(R_{u,j} - \bar{R_u})^2}}$$

Similarities are calculated for every $(i{,}j)$ pair of restaurants found in the data set. $R_{u,i}$ is user u's rating of restaurant i and $R_{u,j}$ is the same user's rating of restaurant j. $\overline{R}_u$ is the average of user u's restaurant ratings. These similarity values are stored in an n by n matrix, where n is the number of restaurants in the data.

3.3 Generating Recommendations

Once we have determined the similarities of all pairs of restaurants, we can predict the user's rating of restaurants he has not yet visited. As displayed in figure 2, we first capture the user's input for restaurants he has visited as well as the ratings he assigns each of these restaurants. Then, we can use a *weighted sum* technique to calculate the user's predicted ranking of a restaurant i by summing the weighted ratings of restaurants similar to i and then dividing this value by the sum of these similarity weightings. Sarwar *et. al* give the following formula to calculate the predicted rating of restaurant i by a user u:

$$P_{u,i} = \frac{\sum_{\text{all similar items, N}}(s_{i,N} * R_{u,N})}{\sum_{\text{all similar items, N}}(|s_{i,N}|)}$$

where s_i is the similarity of an input restaurant to restaurant i and R_u is the user's rating of that restaurant. This calculation is carried out for every restaurant in the area that the user has not visited, and then these restaurants are then sorted by their predicted ratings. The restaurant with the highest predicted rating is the restaurant Mine & Dine deems that the user will enjoy the most, and thus it is output as the recommendation.

4. SYSTEM IMPLEMENTATION

4.1 Data Set

Finding a good blog dataset is one of the chief challenges of implementing this system. The main reason why finding a suitable dataset is so difficult is that the process of

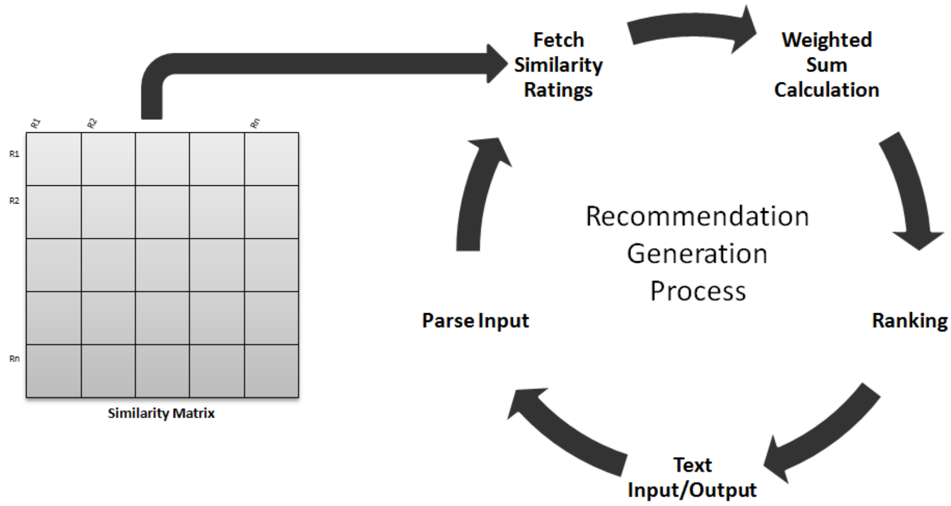

Figure 2: The process of generating recommendations

capturing and indexing blog data is an extraordinarily time-intensive and meticulous process. With millions of blogs in the blogosphere, manually formatting and storing blogs is next to impossible. There are some companies like *Spinn3r* (www.spinn3r.com) that offer real-time indexing of blogs for a fee. The same company also contributed a 142GB XML-formatted blog dataset for the 2009 International Conference on Weblogs and Social Media (ICWSM). An initial implementation of Mine & Dine parsed the *Spinn3r* dataset and used an online web sentiment analysis API called *OpenAmplify* (www.openamplify.com) to determine the polarity of each blog with respect to the restaurant it referenced. In this way, a blog about a restaurant could be turned into a review. Unfortunately, during parsing, not enough blogs referencing restaurants in Philadelphia (the original fixed location for all restaurant recommendations by Mine & Dine) could be found.

Therefore, we decided to use a restaurant review dataset provided by Mehrbod Sharifi and Noemie Elhadad from Carnegie Mellon. This data was collected from a *CitySearch* query of restaurants in New York City. *CitySearch* is a website that indexes and normalizes reviews across the web for venues in different locales across the Unites States. In this way, processing these reviews approximates the effect of processing user opinions on the blog-level of granularity. The data is also XML-formatted. Because this data deals with New York City restaurants, recommendations are restricted to venues in this area.

4.2 Dynamic Web Design

The web interface is arguably one of the most important features in spurring the adoption of such web-based recommendation engines by a wide audience. In the end, if the user doesn't find the interface intuitive, he is not likely to return again for any future venue recommendations. Beyond such aesthetic considerations, however, we have decided to utilize JavaServer Pages (or JSP) to implement the web pages that this system will be housed on. The reason for this choice is the ability that JSP affords the creator to have dynamic content rapidly deployed and easily accessible on any platform or server. Given that there will be a decent amount of user interaction with the interface (up-rating and down-rating recommendations, input of venues, etc.), it makes sense to have JSP in place so that it can accommodate many users as they each interact differently with the system. Java code is used to construct the similarity matrix and to make predicted rating calculations on test data for evaluation purposes.

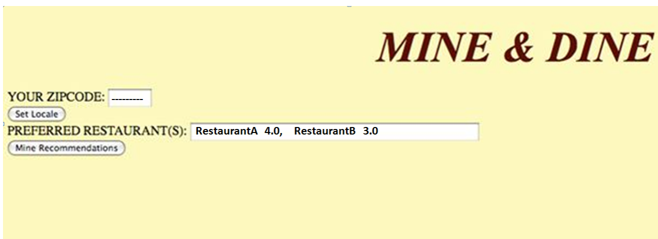

Figure 3: Screenshot of user interface

4.3 Input Capture

The system was designed to capture both the user's location and restaurant-rating data. Because the dataset only deals with New York City restaurants, the location is hard-coded and set to New York City, NY. The format that users must enter their restaurant and rating data in is shown in the following screenshot of the interface (Figure 4). In this example, a user who has visited RestaurantA and RestaurantB rates them as 4.0 and 3.0 respectively.

4.4 Parsing the Data

To parse through the XML-formatted review data, we use *XimpleWare*'s VTD-XML processing model. The document object model (or DOM) constructs trees of nodes in memory and is therefore a very slow option. The SAX processing model is faster than DOM and based on extractive tokens, but it is quite cumbersome to code with. VTD limits its memory usage to 1.3x-1.5x the XML document size and it outperforms SAX in parsing speed. This choice in XML processing model is very important in ensuring good performance. Given the potentially large amounts of review data Mine & Dine must process, the speed of XML parsing becomes the chief bottleneck in constructing the similarity matrix.

We utilize hashmaps to map restaurants to their reviewers and reviewers to their reviewed restaurants. These mappings are necessary for calculating average user ratings and finding subsets of similar users who have rated two restaurants when making similarity calculations. Hashmaps are used for their constant time *get* and *put* functions which makes building and accessing these mappings for similarity calculations relatively quick.

We only include reviewers in the mappings if they have reviewed 10 restaurants or more. The reason for this requirement is to ensure that most of the reviews being processed are coming from users who have rated enough other restaurants to be able to get consistency in similarity calculations between restaurants (a similarity calculation between two restaurants requires users who have reviewed both venues). These user reviews constitute the training data that builds the similarity matrix. Reviews from users who have reviewed more than 4 and less than 10 restaurants are used as test data, because this range of reviews seems like a plausible amount of restaurant ratings that a user would input when using Mine & Dine.

5. EVALUATING MINE & DINE

5.1 Method

The effectiveness of Mine & Dine is dependent upon how well the system predicts the user's ratings of restaurants he has not yet visited. The restaurants are ranked by their predicted ratings, and the highest ranked restaurant is recommended to the user. Because review aggregation websites like *Yelp* do not make explicit recommendations and just rank restaurants by their mean rating, we assume that a user of these sites takes as a recommendation the highest-ranked restaurant shown. Since both Mine & Dine and *Yelp* rank restaurants by ratings, the differentiating factor between the two systems is how well the restaurant ratings match with how the user himself would have rated each venue. Given that *Yelp* rates restaurants based upon their mean ratings, we set the mean restaurant rating "strategy" as the baseline to compare Mine & Dine's collaborative filtering strategy against. The intuition is that collaborative filtering offers a level of personalization of ratings that review-aggregation sites do not, and thus restaurant ratings from Mine & Dine will better match the user's tastes than the baseline.

To quantify the improvement that Mine & Dine brings to recommendations, we segment the restaurant review dataset into training and test sets. The training set is composed of all reviews made by reviewers who had reviewed 10 or more restaurants. The test set is composed of reviews made by reviewers who had reviewed more than 4 and less than 10 restaurants. The test data is stored as a hashmap of users mapping to their restaurant ratings. We calculate the mean absolute error (MAE) and mean absolute percentage error (MAPE) for the two systems' predicted ratings (*Yelp*-style ratings and Mine & Dine-style ratings) of the first restaurant review made by each user in the test set. To test Mine & Dine's rating strategy, we let the second restaurant in each user's set of reviewed venues be input as a restaurant he has visited before. We iterated this process of including more of the restaurants from each user's review sets as input into Mine & Dine. Figure 5 shows this iteration process. The max number of restaurants from each user's review set that we input into Mine & Dine was 8.

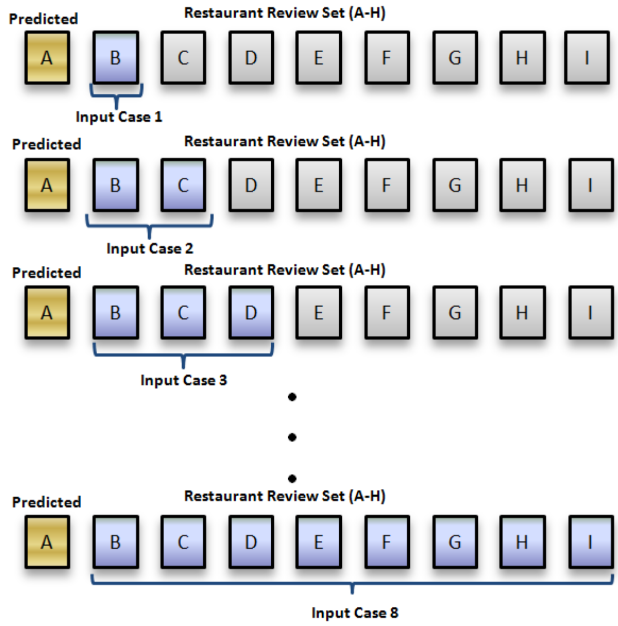

Figure 4: Sources of Input for MAE and MAPE testing

5.2 Results

Figure 5 below shows the results of calculating the MAE and MAPE respectively for the predicted ratings of the specified restaurants in the test data. The leftmost column in both graphs is labeled "Yelp", because the mean rating strategy that *Yelp* implicitly uses is employed to predict the user's predicted rating of a restaurant (equivalent to the mean rating of that restaurant). The other input size cases refer to the number of restaurants pulled from the user's review sets that are used as input to Mine & Dine's collaborative filtering process.

We can see from these results that Mine & Dine's performance at predicting users' ratings gets better with more restaurants that are available as input. This is expected, as more restaurants input to the collaborative filtering process gives the system a better idea of what the user likes. It is not surprising to see a jump in MAE and MAPE when moving from the *Yelp* ratings to Mine & Dine ratings with 1 restaurant input; the system is basing all of its recommendations off of just one piece of information then. With an input of 4 restaurants and their ratings, Mine & Dine performs better than the *Yelp*-style rating system. With an input of 8 restaurants, we see a significant drop in MAE from 0.337 to .231 and in MAPE from 0.896 to 0.716. That is almost a 20% improvement in prediction accuracy over the *Yelp* baseline.

While there is significant improvement in prediction accuracy, the number of past restaurants that need be input is

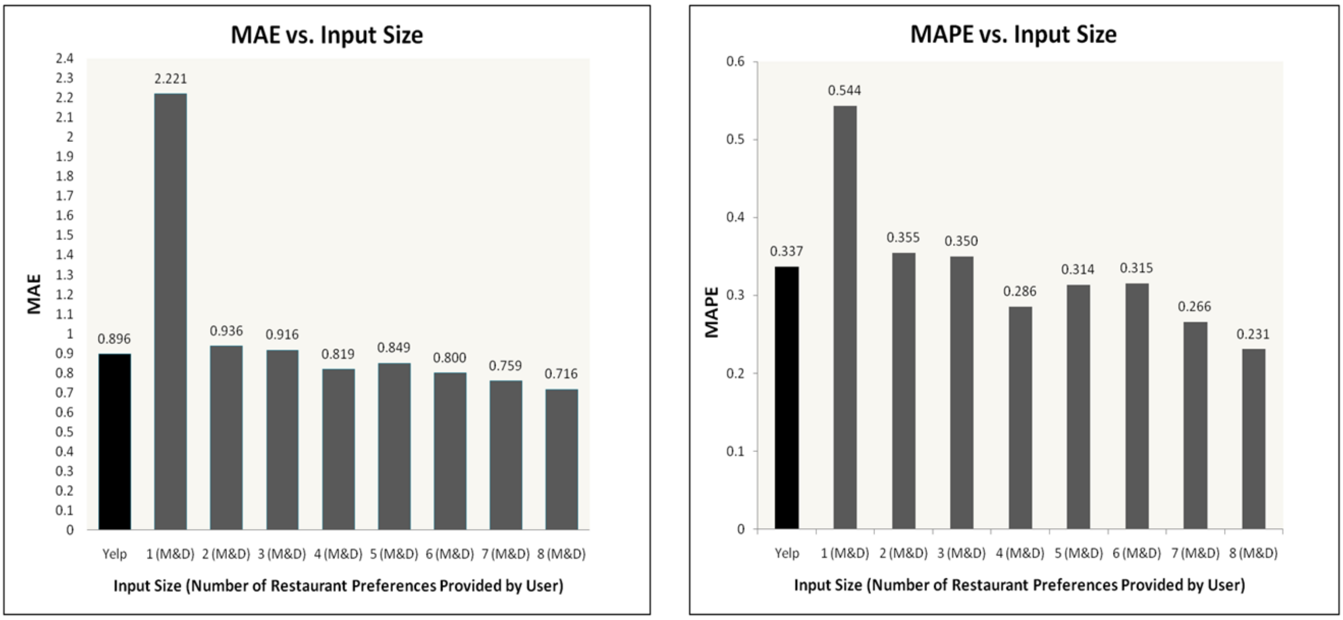

Figure 5: Graphs of MAE and MAPE for recommendations

quite high at 8. Mine & Dine only performs better than the baseline with an input of 4 restaurants or higher. Many users probably expect to have a well-constructed recommendation with just one or two input restaurants, however. Apparently, the weighted sum calculation using similarity scores cannot outperform the wisdom of crowds until enough about the user is known (in terms of restaurant preferences).

6. FUTURE WORK

There are some areas where future work can be done to improve the Mine & Dine system. First, finding a review dataset that pulls reviews from more blogs and social media would only improve the quality of similarity calculations for the different restaurants. If a suitable blog dataset was constructed, sentiment analysis could be conducted on relevant blogs to transform them into reviews- this would help reduce "brag or moan" bias that skews restaurant ratings. Second, Mine & Dine performs best when the user inputs a large number of restaurants that he has visited before. It can become cumbersome and detrimental to the user experience to have to enter in 8 restaurants and their ratings every time a new recommendation is desired. Perhaps, cookies or some membership service can be implemented that stores user information so that recommendations can be refined rather than re-generated. Finally, alongside recommendations, local deals and coupons that are associated with the recommended restaurant could easily be linked to on the web page.

7. CONCLUSION

With Web 2.0 making room for Web 3.0, the age of aggregating data is giving way to the age of processing and utilizing that data. Increasing numbers of consumers are taking to the web in their search for quality recommendations regarding their purchases. The skyrocketing popularity of review-aggregation sites like *Yelp* and *TripAdvisor* is a testament to this emerging market for information. In this paper we presented the Mine & Dine recommendation system that takes a novel approach towards making restaurant recommendations in that it pulls reviews from a broad range of websites and social media, and it uses an item-based collaborative filtering process to personalize recommendations for users. Evaluating how well this system predicts users' ratings of restaurants, we found that Mine & Dine achieved near a 20% reduction in MAPE when compared to the baseline review-aggregation systems. This substanial improvement means that recommendations made through Mine & Dine are more probable to be high quality recommendations.

8. REFERENCES

[1] S. Blair-Goldensohn, K.Hannan, R. McDonald, T.Neylon, G. Reis, and J. Reynar. Building a sentiment summarizer for local service reviews. In *WWW2008 Workshop on NLP Challenges in the Information Explosion Era*, 2008.

[2] K. Dave, S. Lawrence, and D. M. Pennock. Mining the peanut gallery: Opinion extraction and semantic classification of product reviews. In *Proceedings of WWW*, 2003.

[3] A. Esuli and F. Sebastiani. Sentiwordnet: A publicly available lexical resource for opinion mining. In *Proceedings of Language Resources and Evaluation (LREC)*, 2006.

[4] M. Hearst. Direction-based text interpretation as an information access refinement. *Text-Based Intelligent Systems*, 1992.

[5] J. A. Horrigan. Online shopping. *Pew Internet and American Life Project Report*, 2008.

[6] N. Hu, P. A. Pavlou, and J. Zhang. Can online reviews reveal a product's true quality?: Empirical findings and analytical modeling of online word-of-mouth communication. In *Proceedings of Electronic Commerce (EC)*, 2006.

[7] Ryan McDonald and Ivan Titov. A joint model of text and aspect ratings for sentiment summarization. In *Proceedings of ACL-08: HLT*, 2008.

[8] S. Morinaga, K. Yamanishi, K. Tateishi, and T. Fukushima. Mining product reputations on the web. In *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, 2002.

[9] Phillip Nelson. Information and consumer behavior. *Journal of political economy*, 78(2):311, 1970.

[10] Bo Pang and Lillian Lee. A sentimental education: sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, 2004.

[11] Bo Pang and Lillian Lee. Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval*, 2(1-2):1–135, January 2008.

[12] B. Sarwar, G. Karypis, J. Konstan, and J. Reidl. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the World Wide Web Conference*, 2001.

[13] Sylvain Senecal and Jacques Nantel. The influence of online product recommendations on consumers' online choices. *Journal of Retailing*, 80(2):159–169, 2004.

[14] J. Wiebe. The influence of online product recommendations on consumers' online choices. *Computational linguistics*, 20(2):233, 1994.

[15] J. Wiebe and R. Bruce. Tracking point of view in narrative. In *Proceedings of the AAAI Spring Symposium on Empirical Methods in Discourse Interpretation and Generation*, 1995.

[16] T. Wilson, J. Wiebe, and P. Hoffmann. Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of the Human Language Technology Conference and the Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP)*, 2005.