myDeal: A Mobile Shopping Assistant Matching User Preferences to Promotions

Kartik Muralidharan†, Swapna Gottipati†, Narayan Ramasubbu‡, Jing Jiang†, Rajesh Krishna Balan†
†School of Information Systems, Singapore Management University
‡Katz Graduate School of Business, University of Pittsburgh
†{kartikm.2010,swapnag,jingjiang,rajesh}@smu.edu.sg ‡nramasubbu@katz.pitt.edu

ABSTRACT

A common problem in large urban cities is the huge number of retail options available. In response, a number of shopping assistance applications have been created for mobile phones. However, these applications mostly allow users to know where stores are or find promotions on specific items. What is missing is a system that factors in a user's shopping preferences and automatically tells them which stores are of their interest. The key challenge in this system is twofold; 1) building a matching algorithm that can combine user preferences with fairly unstructured deals and store information to generate a final rank ordered list, and 2) designing a mobile application that can capture user preferences and display deal information to the user in an intuitive way. In this paper, we present myDeal, a system that automatically ranks deals according to user preferences and presents them to the user on their mobile device.

1. INTRODUCTION

"Jill is walking down the street and notices a new mall occupying a formerly vacant lot. Similar to other malls in her country, this mall has 100 or more stores inside. Jill loves to shop but the thought of having to walk through this new mall and identify the shops that are of specific interest to her is not that appealing. In addition, even for her popular stores, she really only wants to visit them if they are having a sale or other interesting promotions. She wishes that there was an application on her mobile phone that would automatically tell her of stores and promotions of interest to her everytime she encountered a mall or other shopping area."

The above scenario happens frequently to consumers in large urban Asian cities, such as Tokyo, Bangalore, Hanoi, Singapore, etc., where commercial shopping malls are interspersed with regular office and residential buildings. In addition, in these Asian cities, it is not uncommon for new malls to appear and disappear fairly regularly. The store selection in existing malls also frequently change in response to market pressures and shopping trends. Finally, the density of malls in these cities tend to be quite high (malls can be just tens of meters apart from each other in the worst case).

In this paper we present *ClimbTheWorld*, a smartphone *serious game* aiming at incentivize people to use stairs instead

MOBIQUITOUS 2014, December 02-05, London, Great Britain

DOI 10.4108/icst.mobiquitous.2014.257708

it is not sufficient to just identify interesting stores by name as consumers also want to identify stores that are having deals of specific interest to them [20]. These promotions come in two forms — 1) promotions that are offered by the store regardless of payment implement (50% off storewide sale for example), and 2) promotions that are offered if specific payment or discount implements are used (for example, 20% off when using a VISA card, or mileage points if a frequent flyer card is shown, etc.). Finally, consumers would prefer to find the most interesting stores, with best deals from their perspective, from the ease of their mobile device.

Existing representative mobile advertising i.e., location-based advertising, is limited in effective targeting. This is mainly because it delivers promotions based on just the user's current location without considering any other user context. For example, deals for a nearby restaurant promoting group dining offers do not attract people who are dining alone. Other applications like LiveCompare [9] cater toward on-the-fly price comparisons for mobile users, while retail or card specific applications such as Citi Shopper [8] allow users to browse and view card specific deals. However neither application group shows deals that are independent of any product or payment card.

In this paper, we present the myDeal system that attempts to correct the deficiencies of previous solutions and provide consumers with a context-based mobile shopping application. Here, in addition to factoring the user's current location, we allow the user to specify other context such as *preferences* as well as *payment cards* owned; all important pieces of information needed to improve promotion relevance. The complexity in providing this assistance arises from two main sources: 1) matching and ranking the various promotions and stores with the consumer's cards and preferences to find an "optimal" selection, and 2) capturing user preferences and displaying the results of this ranking process on a small phone screen.

The key challenge in the *ranking algorithm* arises from having to combine both structured (easy to understand numeric discounts ("5% off") etc.) and unstructured (free-form text ("A free teddy bear") etc.) promotion information (1). The algorithm also needs to factor in consumer preferences such as "prefer discounts over free gifts" etc (2). Also, most consumers own multiple payment and discount cards and this multiplies the complexity as each individual card (and combinations of) has its own set of promotions and deals (3). Our final algorithm adapts natural language semantic techniques and combines all three factors to provide users with the most relevant promotions and show those immediately — with the

This research is supported by the National Research Foundation, Prime Minister's Office, Singapore under its IDM Futures Funding Initiative and administered by the Interactive Digital Media Programme Office, Media Development Authority.

remaining promotions hidden away from view until the consumer goes and actively looks for the less interesting promotions.

The additional system challenge arises from having to display hundreds of promotions and store information on a small mobile display without burdening the user. This challenge is the main reason why the ranking algorithm component of myDeal is so vital; as a good algorithm allows us to immediately identify the most relevant promotions. In this paper, we therefore focus on an innovative solution for the matching and ranking aspects.

We evaluated myDeal in two different ways; First, we conducted an in-depth analytical evaluation of the ranking algorithm to understand the accuracy of the algorithm. Second, we ran a user study with 43 undergraduate students to understand the effectiveness of the myDeal system. The user study results show that our preference-based system is more accurate (users can find the most interesting promotions better) and faster to use than two other common system designs (1. all promotions just listed alphabetically, and 2. promotional information hidden under various categories) for these types of shopping applications. Overall, this paper makes the following contributions:

- A matching and scoring algorithm that accurately factors user's preferences when ranking deals and is capable of combining structured and unstructured deal information into a single score.
- An operational system that allows users to effectively specify and find quickly the deals that match their preference on their mobile device.

2. RELATED WORK

At one end we have stores with promotions for their products and services. At the other end are consumers with their needs. Ubiquitous advertising [15] attempts to match the two to create the greatest possible impact on the customers. However, such a service is riddled with many challenges [10, 26] the foremost being ad relevancy.

While a number of researchers [6, 18, 21, 29, 33, 34] have suggested the use of user context to provide a higher degree of ad personalization, most research has focused on using only location to increase the relevancy of advertisements delivered. Alto et al. proposed a location-based advertising (LBA) system that proactively sends ads to mobile phones when a user passes by a certain store, using Bluetooth for localization [1]. In the AdNext system [14], authors propose to use mobility patterns to predict which store the user is likely to visit next, and show him advertisements related to that store. In [12], authors consider using various physical contexts such as location and user activities to serve ads. Commercial LBAs such as Shopkick [31] and PROMO [23] also exploit current user location to push relevant advertisements. More recently SmartAds [22] deliver contextual ads to mobile apps by taking into account the content of the page the ads are displayed on. These works are orthogonal to myDeal; myDeal allows users to *explicitly* state their preferences and *pull* relevant promotions onto their mobile device whenever required. The key focus of myDeal is being able to match these user preferences with fairly unstructured ad content.

Other competitors fall into two broad categories. In the first category, applications like Froogle [11], LiveCompare [9], CompareEverywhere [13] and ShopSavvy [25] cater towards on-the-fly price comparisons for mobile users. In the second category are deal information services, like Citibank's Card Information Service [8] and mobile applications like Mobiqpons [19], that attempt to inform consumers about the best deals available. However, these applications fail in two main areas; first, they are usually targeted to just certain malls and/or certain payment cards (only Citibank cards for example), and second, they do not take into consideration user preferences - factors that are important in getting the best deal! In all the cases we observe that the burden of finding the best deal is left to the consumers — the applications just provide different ways to find lists of deals matching certain broad criteria (such as all restaurant deals etc.).

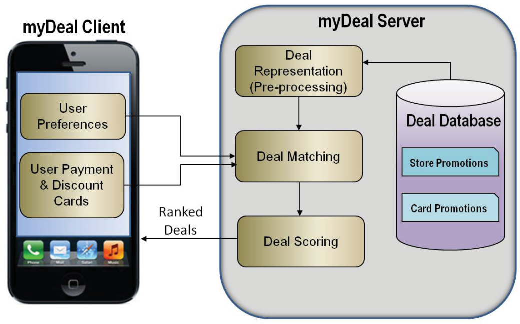


Figure 1: myDeal System Architecture.

Collaborative filtering(CF) techniques have also been used for targeted advertising [7, 27, 30]. CF requires users to actively participate and express his or her preference by rating items in the system. These inputs are used to build user-item (or user-rating) matrices and recommend similar promotions for users with similar taste. We believe however, that user interests are highly situational and hence require myDeal users to explicitly state their preference every time the system is used. However, myDeal could incorporate such additional signals to further increase promotion relevance.

Natural language processing(NLP) techniques are gaining prominence in mobile applications. NLP helps to greatly facilitate access and use of applications, encouraging greater adoption. Siri [32], a personal assistant mobile application uses NLP to answer questions and make recommendations. The application adapts to a user's individual preferences over time and personalizes results, and performs tasks such as making dinner reservations while trying to catch a cab. Maluuba designed an API based on NLP and other machine learning techniques that aids in processing the natural human language [17]. In our model, we applied NLP techniques for estimating the semantic similarity of deal items.

3. THE myDeal SHOPPING ASSISTANT

Our system architecture is shown in Figure 1. We first explain our design decisions followed by the solution details of each component of the system.

3.1 Design Considerations

3.1.1 *Factor in User Preferences*

Understanding promotion details is not enough — we also have to factor in the consumer's preferences. Does she want immediate discounts over vouchers? or does she prefer frequent flyer points over everything else?

A major limitation with all existing shopping assistance programs (that we know off) is that they do not really take into account user preferences.At best, the user is allowed to specify categories or keywords and the programs sort the results based on that. However, a single mall can have hundreds of stores with many hundreds of deals between them. As such, even keyword searches and categories break down in this type of rich data environment (we validate this claim in our user study presented later). This problem is made worse when we consider that there are multiple malls usually within walking distance and that the consumer has multiple payment implements (which usually offer promotions on top of those already offered by the stores themselves). A pre-study demographics survey showed that several users carried between 4-6 cards in order to avail such offers.

A key consideration for us was to therefore integrate user preference into our system so that only the most relevant promotions and stores are brought to the consumer's immediate attention. However, this is a non-trivial process as promotions are stated using a combination of structured (easy to understand numeric discounts. e.g."5% off") and unstructured (free-form text. e.g."Free teddy bear") components. A naive approach would be to just rank promotions based on the easy to sort structured components. This is not good as many promotion details, of high interest to the consumer, tend to be located in the unstructured components.

3.1.2 Has to be a Mobile Application

A fast growing smart phone market suggests that consumers really want to have this capability on their mobile phones allowing them to plan their shopping experience anytime and anywhere they wanted to.

However, the smart phone is not without its limitations; chief of which is their small display screen (at most 4-5 inches). This makes it important to design an application that only shows users the promotions and stores that are of highest value to them (hiding the rest away for the user to browse through manually if so desired). This is particularly important when we factor in the hundreds of possible store/deal/card combinations that could be applicable to any given consumer.

3.2 Building myDeal

Keeping in mind the design considerations discussed in Section3.1, building myDeal requires the following three components:

An electronic representation of deals: Currently, most deals are not stored in an electronic form. Our first task(*Deal Representation*) was thus to devise a schema that could capture both stores deals and promotions as well as deals and promotions offered by the various payment and discount cards from *Deal Database*.

Finding top deals: Given a set of cards carried by the user, the user's preferences, and the deals offered by retailers and card issuers, *Deal Matching* and *Scoring* components find the "best deals" for that user.

Presenting deals to the user: Ultimately, it is the user who has to decide which deal is the best for them. Hence, it is crucial that relevant information is presented to the user in a way that makes it easy for them to find the deal that maximizes their needs. In addition, users must also be able to specify their shopping preferences easily on the *myDeal* mobile application.

3.3 Representation of Deals

Deals are offered by two main entities; retailers and payment/discount card operators. In the retailers case, the deal is likely to be valid only at that specific retailer whereas payment/discount card deals are likely to be valid across many retailers. For example, a supermarket could offer a 50% deal on laundry detergent. That deal is likely to only be valid at that supermarket and possibly its branches. On the other hand, a bank could offer a 2% cash rebate on its premium VISA credit card on *all* purchases. This cash rebate would apply no matter where the detergent was bought. Note: it is also possible for deals to be constrained to a particular retailer and a particular card. Our schema, described below, can handle this case as well.

We manually inspected a few hundred deals, from various retailers and payment cards, and identified just four components that were used to create all deals - every deal we encountered was some combination of these four components with different values and attributes assigned to each component.. The four components are:

1. *Cash back*: These are specific cash refund. For example, 3% cash back of entire bill. These discounts are in the form of percentages or fixed values ($5 cash back).

2. *Discounts*: These are specific cash discounts. For example, 5% cash discount of entire bill. These discounts are in the form of percentages or fixed values ($10 discount).

3. *Vouchers*: These are vouchers that can be accrued and then exchanged later for either cash or products. Frequent flyer miles, store loyalty points, etc. are some examples.

4. *Rewards*: These are deals in the form of real products. For example, get a teddy bear free with every $10 purchase.

We use an XML-based schema to describe deals comprising of these four components. In addition to capturing basic deal information, a *Stackable* tag specifies whether this deal can be combined with other deals. For example, a deal offered by a loyalty card may only be usable with a deal offered by a credit card issued by a specific bank.

3.4 Finding the Best Deal

The second part of myDeal is a matching and scoring subsystem that rank orders the best deals available for the user. These ranks determine how good a deal the user would get if they shopped at a particular retailer, possibly for a particular product, using particular payment and discount cards. Rank ordering the best deals involves two major steps: 1) Matching deals preferred by the user to those offered by the retailer and card issuers and 2) assigning scores for each of these valid combinations.

3.4.1 Matching algorithm

The matching algorithm is dependent on the parameters of four entities involved in most shopping scenarios; namely the deals, the cards carried by the user, user preferences and location. The four entities are mapped in two steps. First, users are mapped to retailers based on their location and the kind of deals they are looking for (e.g. Dining) and their deal preference. Second, retailers are mapped to the cards carried by the user. The matching algorithm filters out those retail outlets that do not match the above criteria.

3.4.2 Scoring algorithm

Deals are generally a combination of structured and unstructured content. Those that only consist of simple numerical values can be scored easily and ranked. The real challenge however is in how we score the following type of deals:

- Deals with multiple numeric values (e.g. Get a complimentary S$30 gift voucher and additional 3% rebate).
- Deals with non-numeric values(e.g. 1-for-1 free lunch)
- Deals with multiple non-numeric values.
- Deals with numeric and non-numeric values.
- Deals with multiple numeric and non-numeric values.

Expt Code	Description	Effect Studied
Single	Going for lunch alone. No choice of cuisine.	Ability to identify the most rewarding deal.
Couple	Going for lunch with a friend. No choice of cuisine.	Ability to identify the most rewarding deal for a couple.
S-Focus	Going for lunch alone. Choice of cuisine is "‘Western"’. Looking for deals offering rewards.	Ability to identify the most rewarding deal with semi-focused options.
V-Focus	Going for lunch in a group. Choice of cuisine is "Chinese"’. Looking for deals offering discount and voucher and prefer voucher value over discount.	Ability to identify most rewarding deal when the social setting is a group, with very focused options.

Table 1: myDeal User Study Experiments

In order to score any deal, we first need to extract values for the following four categories from the deal description: *Discount*, *CashBack*, *Voucher* and *Reward*. For each category we use regular expressions to match and extract the corresponding values. For example, if the deal description is “Get a 20% Discount on the Total Bill and enjoy a complimentary Voucher of $10” we extract the following values *Discount* = 20 and *Voucher* = 10.

We then use the following formula to score a deal:

$$Score = \alpha \cdot Discount \ + \ \beta \cdot CashBack \ + \ \gamma \cdot Voucher + \ \theta \cdot Reward,$$

where α, β, γ and θ are weights to adjust the importance of each deal category. The value of these weights are specified by the user as deal preferences.

3.4.3 Deriving a value for rewards

Consider the following deal description: “Enjoy a Free Ice Cream or Cake with every meal purchased”. To apply a score to this deal we must effectively assign a value to both reward items *Ice Cream* and *Cake*. We propose a 2-step machine learning algorithm that uses semantics to determine the corresponding value of the reward.

Step 1: Get a list of all reward items and their corresponding value (if specified) from all deals. This is done using standard NLP techniques such as part of speech (POS) tagging. For our system we use the Brill POS tagger from CST [5]. In our example, the rewards *Ice Cream* and *Cake* are extracted and added to the list.

Step 2: Using a semantic similarity method described by Lin et. al [16] and Pedersen et. al [24], we cluster rewards in the list into a semantic space in which rewards that are closely associated are placed in the same cluster. Several existing algorithms compute relatedness only by traversing the hypernymy taxonomy and find that *Ice Cream* and *Cake* are relatively unrelated. However, WordNet provides other types of semantic links in addition to hypernymy, such as meronymy (part/whole relationships), antonymy, and verb entailment, as well as implicit links defined by overlap in the text of definitional glosses. These links can provide valuable relatedness information. If we assume that relatedness is transitive across a wide variety of such links, then it is natural to follow paths such as ice cream-frozen dessert-dessert, sweet-dessert and find a higher degree of relatedness between *Ice Cream* and *Cake*.

Lin’s similarity measure uses the information content (IC) of the words/concepts, and the least common subsumer (LCS) of the concepts in the WordNet taxonomy. LCS is the common ancestor of two concepts which has the maximum information content. The similarity measure between concepts w_i, w_j is defined as follows.

$$Sim(w_i, w_j) = \frac{2 \times IC(LCS(w_i, w_j))}{IC(w_i) + IC(w_j)}$$

where

$$IC(c) = -log(P(c))$$

$LCS(w_i, w_j)$ is a common subsumer of w_i, w_j , $IC(c)$ is the information content of the concept c and $P(c)$ is the probability of c. In our example, the rewards *Ice Cream* and *Cake* are likely to

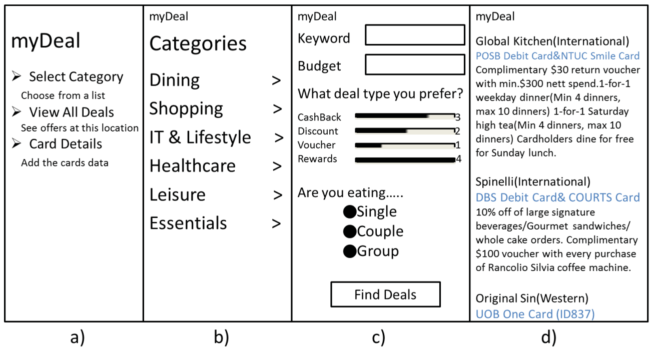


Figure 2: myDeal Usage Sequence on the Window Phone

be clustered together as *’Dessert’*. The median value of the cluster is then assigned to rewards that currently do not have any value associated with them. Going ahead with our example, if the median value of the cluster *’Dessert’* is $5, the rewards *Ice Cream* and *Cake* are also assigned the same value.

3.4.4 Where to perform the match and scoring

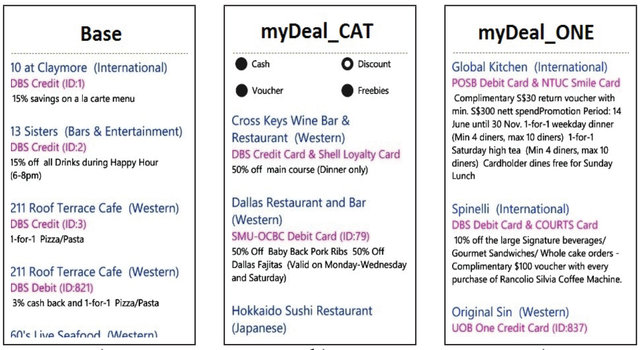
One question we faced when building myDeal was deciding where to perform the ranking. In particular, we could perform the matching and scoring on the user’s mobile phone or perform the ranking using an external service. Each of these options had its strengths and weaknesses.

Rank ordering on the user’s mobile phone provides the highest amount of privacy — no card information is sent anywhere. However, the user’s mobile phone is computationally limited and requires access to deal information across multiple retailers and card issuers. While we are concerned about resource utilization and privacy, as these issues are not the primary focus of our paper, we decided to use a backend service to perform the filtering and scoring, at the risk of users card information(only card name) revealed to the hosting site.

3.5 Integrating the User

The final component of our system is the user interface for presenting deals to users. We designed it to satisfy the following key properties:

- **Clear indication of combination of cards**: Many deals require combining multiple cards together to attain them. It is thus crucial to point out to the user which cards need to be used. We achieved this by using colour coding to distinguish the different pieces of information.
- **Display deal breakdowns**: We achieved this by showing the complete description of discount percentage, cash back, vouchers and rewards.
- **Ordering/positioning deals**: We achieved this by applying a score to each deal and displaying deals in order of this score.



a) b) c)

Figure 3: System Variants

3.6 End-to-End System

Our system works as follows: The user enters a shopping mall and loads the myDeal application (Figure 2a). myDeal presents several options that the user can navigate through. The user can either choose to view all deals offered at that location or may choose to specify a particular category of interest (e.g.Dining, Healthcare, etc.) as shown in (Figure 2b). The user then proceeds to input additional optional details like type of deal preferred, keywords if any, and desired amount to spend and so on as shown in (Figure 2c).

myDeal will extract user's card details from the secure storage area of the phone, append any additional user input and location, and send them to the backend service over an existing wireless communication channel. The ranking service will compute a score which are then ordered and sent to the user's mobile application (Figure 2d).

4. VALIDATION PLAN

In this section, we describe the validation approach used to evaluate myDeal.

4.1 Success Criteria

The goal of our validation was to test if myDeal was successful at presenting appropriate deal information to end users in an easy to use manner. To focus our validation, we identified the following criteria as being crucial for myDeal's success:

1. The scoring algorithm is accurate.
2. Users can make deal decisions quickly.
3. Users can find the best deal accurately.

4.2 Dataset

Our dataset for the algorithm evaluation consisted of real-world dining related deals manually extracted from multiple sources (10 shopping malls and 4 major credit card providers). We used a total of 842 deals from 610 restaurant covering 7 cuisine types in the user study. Also, we observed from a detailed breakdown of deal component combinations (cash back, discount, vouchers and rewards) 5% of the deals offered cash deals while 25% of the deals offered rewards— validating our decision to specifically handle unstructured free-form rewards in our ranking algorithm.

4.3 System Variants

In order to effectively validate our system we introduce three variants. The first variant (Base) displays the deals alphabetically (Figure 3a) by the retail name. An exact keyword search option is also provided. This baseline is representative of the options available with current state of art applications such as CitiShopper [8] and Mobiqpons [19].

The second variant (myDeal_CAT) builds on top of the first and allows user to view deals categorically by the deal components (e.g view deals that offer discounts, deals that offer vouchers etc.) (Figure 3b).Further, within each category deals are sorted according to the numerical value of the deal using only one particular deal component (e.g deals offering 50% discounts will appear higher then those offering 20%). Note that in the myDeal_CAT system we do not calculate aggregate deal score of any kind. Deals that do not include any numerical value would appear below those that do. This variant represents a natural transitional progression between the baseline and the full myDeal system.

The third variant (myDeal_ONE) displays deals (Figure 3c) ordered by the aggregate deal score calculated by our algorithm. In addition to a keyword search option users can also input their preferences on the deal components they prefer (e.g., they prefer discounts twice as much as vouchers etc.) as well as provide other information such as their budget.

4.4 Experimental Procedure

We recruited a total of 43 undergraduate students for our tests. The participants worked alone in a lab for the duration of the study. They were provided a Windows Phone containing all three of our system variants. Each participant was then given the instructions for our study and provided with basic training in how to use the phone. The training period lasted for at most 5 minutes, and consisted of having the participant start the baseline application on the phone and understand the various navigation options.

All 43 students completed the same set of tasks (shown in Table 1). Their goal in all tasks was to select the best deal in terms of the overall savings achievable. We told the participants that they should take into account each component of the deal when calculating the overall savings possible for that deal. The procedure for each task was as follows; first, the participant was given a scenario that they had to follow (e.g., you are eating lunch alone and feel like having chinese food and perhaps an ice-cream cone afterwards). They were then provided with deal information (for several cards and retailers) and were asked to pick the best deal in their opinion. They were free to select any deal (we noted down their final selection) and they were not allowed to ask for any help in the selection process. Also, the overall deal scores computed by our algorithm for each deal were not available to the user. We intentionally omitted the overall score as we did not want to bias the user's perception. Instead, we wanted to test if the deals that the users thought were the best matched what our algorithm considered to be the best.

During each task, we observed the time taken for selecting a deal for the presented scenario. We emphasised that they were not under any time pressure, and could take as long as they needed to complete the task. This was a deliberate bias against our goal of fast transaction times. We first performed all the experiments on the base system and then followed up with myDeal_CAT and myDeal_ONE. We minimised any learning effect by randomising between myDeal_CAT and myDeal_ONE. After each task was completed, participants were presented with a brief questionnaire (that used an easy 5-point Likert scale) that captured their perceived ease of use and accuracy of the completed task. After completing all tasks, users filled an end of experiment questionnaire.

Note that in a real scenario, the number of deals presented to the user will be significantly less than that displayed for each of the above test scenarios. Recall that in a real scenario deals are first filtered based on location and would typically correspond to those

Scenario	Weights (%)				Rank	Error
No.	C	D	V	R	Diff.	Mag.
1	100	0	0	0	0.0 (0.0)	0.0 (0.0)
2	0	100	0	0	1.2 (1.03)	2.0 (2.0)
3	0	0	100	0	0.1 (0.0)	1.0 (0.0)
4	33.3	33.3	33.3	0	1.0 (1.05)	2.0 (0.0)
5	50	50	0	0	1.0 (1.05)	2.0 (0.0)
6	50	0	50	0	0.0 (0.0)	0.0 (0.0)
7	0	50	50	0	0.5 (0.7)	1.7 (0.5)
8	0	0	0	100	2.5 (2.0)	3.6 (1.3)
9	25	25	25	25	1.6 (1.26)	2.0 (1.07)
10	25	25	0	50	1.8 (1.8)	2.6 (1.6)
11	25	0	25	50	5.1 (4.2)	5.1 (4.2)
12	0	25	25	50	1.0 (1.4)	2.0 (1.4)

C=Cash Back, D=Discount, V=Vouchers, R=Rewards. Values in parentheses are standard deviations.

Table 2: Accuracy of Algorithm Relative to Expert

available in a single mall. For our study however, we deliberately chose to ignore location and present the participant with *all* deals that matched the test scenario. This was done to increase task complexity and test our ranking algorithm for a larger subset of deals. Ignoring location also allows us to effectively benchmark our system against applications (such as the Base variant described earlier) that show all available deals irrespective of location.

5. EXPERIMENTAL RESULTS

In this section, we present the results of our evaluation. The goal of the evaluation was to determine if myDeal satisfied the success criteria.

5.1 Results: The Algorithm is Accurate

We evaluated the ranking algorithm by comparing its scores for the top 10 deals versus the scores of 3 experts using twelve different scenarios. In each scenario, we changed the weights allocated to each of the four deal components. Table 2 shows the average difference in rank position and the average error magnitude (for all the errors that were made, what was the average error) for the top 10 deals ordered by the experts with that of our algorithm for all twelve scenarios. The total time taken by our algorithm to score all 842 deals used in the user study was 312ms - well within reasonable limits.

The values in the "Rank Difference" columns indicate the average difference in the score ranks assigned by each entity for each set of deals while the numbers in the second column show that in the event of a rank variation what the average difference would. A lower number in both columns indicate a higher level of agreement between the two entities in terms of scoring.

The first seven scenarios were evaluated without considering the unstructured data part of the deal (Rewards in the form of free text). The results show that the algorithm was effectively able to extract values from the structured data part of the deal. In particular, the largest average difference between the experts and the algorithm was just 1.2 (scenario 2) and even in that case, the average magnitude of the error was just 2 scoring positions (i.e., if the expert ranked a deal 3rd, if the algorithm made a mistake, it would, on average, rank that deal 1st or 5th).

In the last 5 scenarios (scenarios 8 to 12), the deal scores included the free-form text of the deal. The average rank difference and error magnitude in this case is relatively higher. This result was expected as scoring would involve assigning a value to the free text which is subjective. The algorithm uses similarity matching to assign this value while the expert uses heuristics to do the same. Overall however, the average difference is not large enough to invalidate the accuracy of our algorithm.

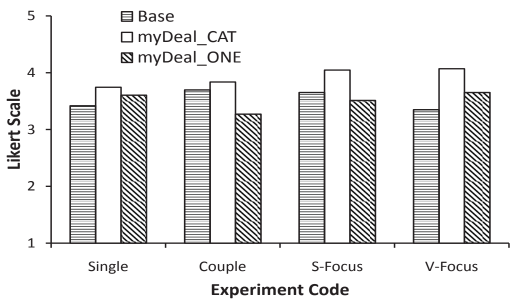


Figure 4: Is myDeal Perceived to be Easy to Use?

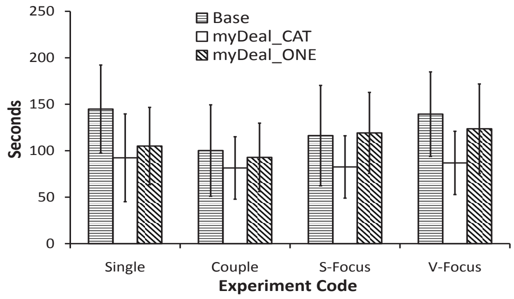


Figure 5: Measured Task Time.

5.2 Results: myDeal is Easy to Use

Figure 4 shows the perceived ease of use of the two myDeal variants and the baseline system. The graph shows the averages of the self-reported Likert score. From the Figure, we see that all three systems are perceived to be easy to use with myDeal_CAT being slightly better than the other two. This is perhaps indicative from the fact that all participants were undergraduate students who are quite mobile savvy and thus quite likely to be comfortable using these types of applications.

5.3 Results: myDeal is Fast to Use

Figure 5 shows the measured times taken for users to finish each experiment. We observe that the myDeal_CAT times are always lower than the corresponding times taken for the other two systems; with myDeal_ONE being only slightly lower than the baseline system. The lack of significant time differences between the baseline system and the myDeal_ONE variant can be explained by the time needed for users to input additional information such as deal preferences in myDeal_ONE.

5.4 Results: myDeal is Accurate

We measured the accuracy of each system by comparing the score of the deal chosen by each user for each experiment. Figure 6 shows the results of that comparison. These results show that user perception and reality can be very different.

In particular, we see that the the average deal scores of myDeal_ONE are significantly higher than myDeal_CAT and the baseline in certain experiments — indicating that the users obtained

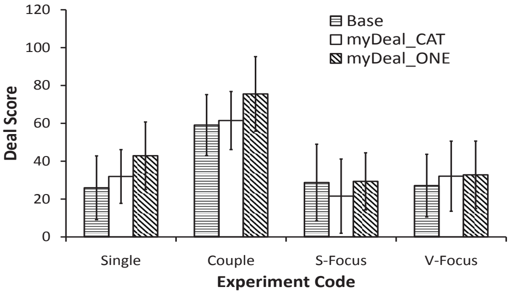


Figure 6: Measured Accuracy.

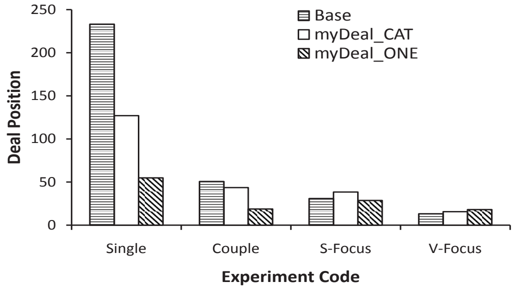


Figure 8: Absolute Positioning of Deals.

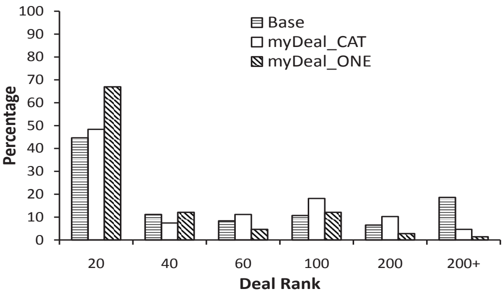


Figure 7: Rank Distribution of Deals.

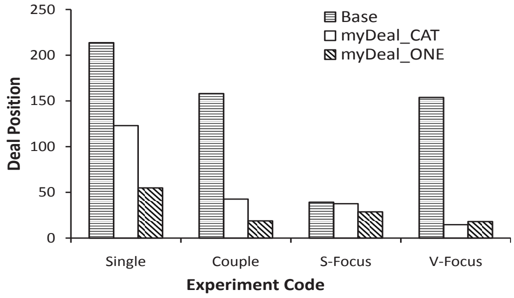


Figure 9: Relative Positioning of Deals.

significantly better deals (higher value overall deals) using myDeal_ONE than the other two systems.

Note that the absolute accuracy of myDeal_ONE is low in certain experiments (S-Focus and V-Focus) as a few users chose alternate, poor scoring deals, to the best deal shown on the mobile device for reasons unknown to us. This is further shown by the histogram in Figure 7 that shows that almost 70% of the deals chosen in myDeal_ONE were within the top 20 deals overall. The lower than expected accuracy was due to just a few tragically bad choices where the chosen deals were more than 200 positions away from the best deal.

5.5 Results: Best Deal is on Top

A key decision we took was not to display the deal score in myDeal_ONE. This was to prevent any bias in the selection of deals. As the user was free to select any deal that matched the given experiment scenario, the position of the selected deal is compared to the top deals chosen by our algorithm. This provided the second phase evaluation of the algorithm.

Figure 8 shows the absolute difference in position of the deal selected by the participant from from the top ranked deal for that experiment and Figure 9 shows the relative distance between the deal selected by the user and the top ranked deal as displayed on the screen. Clearly most users selected the deals closest to the top most deal (within top 20) in myDeal_ONE.

We analysed our results to understand why most users were only able to pick the 20^{th} or so top deal relative to our algorithm's top choice and identified two main factors; First the users did not really know how to extract the various components of the deal and assign scores to them. Many users were assigning values for Vouchers to Rewards for example. This was a result of our intentional decision to not train users in deal ranking methods and to not show the overall score of myDeal. Second, users frequently picked deals that they thought were good even if the deal itself was not that good. This was to be expected as without an objective guide (like our overall score), humans are quite likely to go with a "gut" feel over what seems to be better. We believe that each of these issues will be corrected when using the full version of myDeal that shows the actual scored ranks with a detailed breakdown of the score components.

5.6 Summary of Results

Overall, the user study demonstrated that users preferred a combination of the myDeal_ONE and myDeal_CAT UI variants to obtain the best deal information. This is reinforced by the self-reported user scores, shown in Figure 10, where every user indicated that both myDeal_ONE and myDeal_CAT were useful systems to have – with most users having an overall positive opinion of myDeal_ONE. We also validated the algorithm accuracy by showing that most users tend to select deals that were within the top 20 deals overall when using myDeal_ONE.

6. DISCUSSION

6.1 Time Pressure

In addition to the experiments listed in Table 1 we asked each participant to repeat the baseline experiment "Single". However, this time, they were given a hard time limit of 1 minute. This was to test our system variants in a real-world situation where users do not have much time to look for deals. Figure 11 shows the average score of the deals selected by the participants for each UI variant under this time pressure. It is clear that myDeal_ONE works very well under a time constrained scenario as compared to the other two variants.

Figure 12 validates this in terms of the absolute and relative position of the selected deal as compared with the absolute best deal. Figure 12 shows that deals selected using myDeal_ONE in a time constrained environment are far more likely to be closer to the best possible deal as compared to the other two systems.

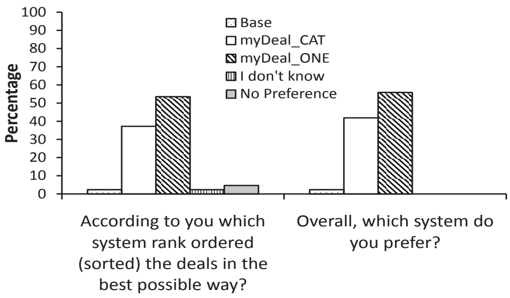


Figure 10: Overall Usefulness of myDeal.

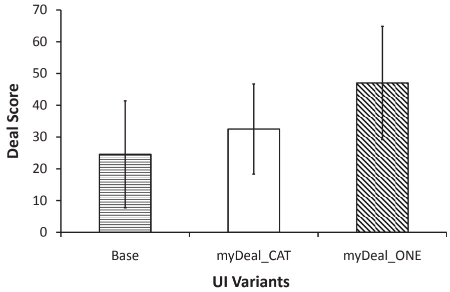


Figure 11: Accuracy in a Time Constrained Scenario

6.2 Limitations of the Ranking Algorithm

A key part of the algorithm complexity lies in extracting both the structured and unstructured information parts of the deal. As description of deals are not standard, extraction algorithms are limited by the current state-of-the-art in natural language processing rules and technologies. In our current prototype, we currently cannot handle deals that include elaborate free-form conditions to satisfy the deal requirement. For example, several deals require purchasing additional items in order to redeem the primary offer. Any additional purchase should effectively reduce the overall score and the algorithm would need to determine the value of this additional item and reduce the overall score by the appropriate factor. Our system currently cannot handle these types of deals in a general way.

Secondly, there are several instances where deals are assigned the exact same score. This is obvious as payments card issuers and retailers tend to offer similar deals through a joint promotion (similar to code-sharing in airlines). In this case, deals within the same score cluster are listed alphabetically. This ordering could perhaps bias the user. We propose in subsequent versions to collapse deals with the same score in a manner that would allow the user to make a more informed decision.

6.3 Real World Deployment Issues

Deployment of myDeal in real world situations is likely to face several challenges. First, a successful deployment of myDeal depends on the extent to which issuers of payment and reward cards along with retailers are willing to share accurate information about their individual promotions. Because myDeal is designed to reduce end user deal searching costs, it is likely that competing agencies might withhold sensitive information rather than willingly share them; creating an operational hindrance for myDeal — an effect observed by previous empirical studies [3]. Next, there is a need for efficient dispute resolution mechanisms between users and retailers when there is a mismatch in the deal information they possess. Such overheads are likely to be a deterrent for the deployment of myDeal.

On the other hand, myDeal can also have potential positive effects on businesses. For example, the positive experience of getting the best deal might boost consumer's satisfaction – possibly resulting in repeat purchases. Similar to the use of other online recommendation systems [2, 28], retailers can utilise myDeal, to improve customer loyalty and differentiate themselves in the market place through customer empowerment.

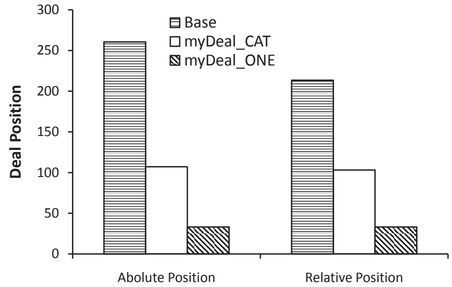


Figure 12: Deal Position — Time Constrained Scenario

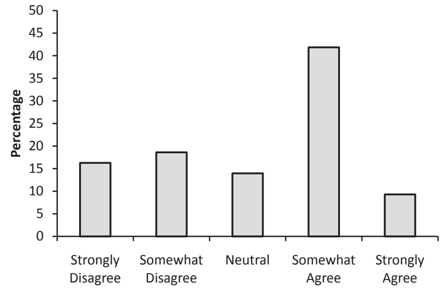


Figure 13: Will Users Share Personal Information?

6.4 Improving our System

We intentionally chose not to show the final deal value (as scored by our ranking algorithm) to the participants in the user study. This was an intentional bias against our system as we did not want to influence the user's final deal choice (in ways other than the position of the deal in the system variants). The omission of the final score was particularly noticeable when free-text deal components had to be accounted for as different users assigned very different values to these free-text elements (many users reported not having a basis to assign the values; hence they just made up something). However, the final deployed version of myDeal will present the score values and we believe that will make myDeal significantly better in practice than the results obtained from the user study. Our final version will also allow the user to see the detailed breakdown of the final score to determine which parts of the score was provide by which components of the deal in question.

Also, our current prototype only asks users for their weights in relation to the four deal components used in our algorithm (Cash Back, etc.). However, it is possible to improve the accuracy of the algorithm if the user is willing to provide additional information such as their past shopping history, club memberships, frequent flyer preferences, etc. At the end of our user study, we asked each user if they would be willing to provide additional personal information if the deals they received were better for them. The results [Figure 13]of the survey indicated that most users were willing as long as they obtained tangible benefits. However, this additional information must be balanced with the need to build a simple to use interface that does not require the user to spend a long time configuring their choices. We plan to investigate this tradeoff between a rich set of user preferences and a simple interface design in the near future.

Finally, in addition to making the current application market-ready, we are also building Android and iPhone versions of myDeal. This work is part of a larger research initiative called *LiveLabs* [4], that hopes to gain an understanding of mobile usage patterns with real users, using their regular phones in real-world environments. In doing so we aim to build new pervasive retail applications and services, such as context-based dynamic promotions and social/group purchasing recommendations.

6.5 Other Limitations

This paper has a number of limitations that we are aware of. First, the card descriptions have only been validated qualitatively with a number of real-world cards. It is quite possible that we will need to edit the schema to support deals from a previously unknown real card. However, we feel that the schema is quite complete and is capable of handling most of the deals available today. We plan to continue testing (and editing where necessary) our schema with real cards.

Finally, the user study used 43 undergraduates in a controlled lab environment. This leads to a clear bias as a) the sample size is small from a social science perspective, and b) undergraduates tend to be more tech-savvy than the general older population. However, we feel that the results will still be fairly indicative of a large slice of the shopping public. In addition, as the study was performed in a controlled environment it is possible that a real field study will generate different trends and results.

7. CONCLUSION

A common conundrum in pervasive computing is deciding how to present information to users of these systems. On one hand, users should be provided as much information as possible so that they can make better decisions. However, providing this much information tends to overload the user and have negative usability impacts.

A common technique to reduce this information overload is to use automation (in the form of AI or similar techniques) to process the data and then provide the user with only the pertinent information.

In this work, we attempted to find the sweet spot between automation and user intervention in the specific context of finding the best deal that matched the user's criterion. We created myDeal, a system that automatically ranked deals according to user preferences and presented to the user in an efficient way on their mobile device. The results of the user study were very promising and showed that users liked myDeal and were more accurate in picking the best deal when using our system.

8. REFERENCES

[1] L. Aalto, N. Göthlin, J. Korhonen, and T. Ojala. Bluetooth and wap push based location-aware mobile advertising system. In *Proceedings of the 2nd international conference on Mobile systems, applications, and services*, MobiSys '04, pages 49–58, New York, NY, USA, 2004. ACM.

[2] A. Ansari, S. Essegaier, and R. Kohli. Internet recommendation systems. *Journal of Marketing Research*, XXXVII:363–375, Aug. 2000.

[3] J. Y. Bakos. Reducing buyer search costs: Implications for electronic marketplaces. *Management Science*, 43(12):1676–1692, Dec. 1997.

[4] R. K. Balan, A. Misra, and Y. Lee. Livelabs: Building an in-situ real-time mobile experimentation testbed. In *Proceedings of the 15th Workshop on Mobile Computing Systems and Applications*, HotMobile '14, pages 14:1–14:6, New York, NY, USA, 2014. ACM.

[5] Center For Sprogteknologi. *POS-tagger*. `http://ida.hum.ku.dk/tools/index.php/?lang=en`.

[6] P.-T. Chen and H.-P. Hsieh. Personalized mobile advertising: Its key attributes, trends, and social impact. *Technological Forecasting and Social Change*, 79(3):543–557, Mar. 2012.

[7] E. N. Cinicioglu, P. P. Shenoy, and C. Kocabasoglu. Use of radio frequency identification for targeted advertising: A collaborative filtering approach using bayesian networks. In *Proceedings of the 9th European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty*, ECSQARU '07, pages 889–900, Berlin, Heidelberg, 2007. Springer-Verlag.

[8] Citibank Singpore Ltd. *Citi Shopper (SM)*. `http://citishopper.citi.com/`.

[9] L. Deng and L. P. Cox. LiveCompare: grocery bargain hunting through participatory sensing. In *Proceedings of the 10th Workshop on Mobile Computing Systems and Applications (HotMobile)*, Santa Cruz, CA, Feb. 2009.

[10] C. I. Eriksson and M. Akesson. Ubiquitous advertising challenges. In *Proceedings of the 2008 7th International Conference on Mobile Business*, ICMB '08, pages 9–18, Washington, DC, USA, 2008. IEEE Computer Society.

[11] Google Limited. *Froogle*. `http://www.google.com/products`.

[12] M. Hardt and S. Nath. Privacy-aware personalization for mobile advertising. In *Proceedings of the 2012 ACM conference on Computer and communications security*, CCS '12, pages 662–673, New York, NY, USA, 2012. ACM.

[13] J. Sharkey. *CompareEverywhere*. `http://compare-everywhere.com`.

[14] B. Kim, J.-Y. Ha, S. Lee, S. Kang, Y. Lee, Y. Rhee, L. Nachman, and J. Song. Adnext: a visit-pattern-aware mobile advertising system for urban commercial complexes. In *Proceedings of the 12th Workshop on Mobile Computing Systems and Applications*, HotMobile '11, pages 7–12, New York, NY, USA, 2011. ACM.

[15] J. Krumm. Ubiquitous advertising: The killer application for the 21st century. *IEEE Pervasive Computing*, 10(1):66–73, Jan. 2011.

[16] D. Lin. An information-theoretic definition of similarity. In *ICML*. Morgan Kaufmann, 1998.

[17] Maluuba Personal Assistant App. `http://techcrunch.com/2012/11/14/maluuba-launches-natural-language-processing-api-brings-siri-like-powers-to-any-app/`.

[18] M. Merisavo, S. Kajalo, H. Karjaluoto, V. Virtanen, S. Salmenkivi, M. Raulas, and M. Leppäniemi. An empirical study of the drivers of consumer acceptance of mobile advertising. *Journal of Interactive Advertising*, 7(2):41–50, 2007.

[19] Mobiqpons. *Mobile Coupons*. `http://www.mobiqpons.com/`.

[20] S. Moriarty, N. Mitchell, and W. Wells. *Advertising: Principles and Practices*. Upper Saddle, New Jersey: Pearson Education, Canada, Ltd, 2009.

[21] C. Narayanaswami, D. Coffman, M. C. Lee, Y. S. Moon, J. H. Han, H. K. Jang, S. McFaddin, Y. S. Paik, J. H. Kim,

J. K. Lee, J. W. Park, and D. Soroker. Pervasive symbiotic advertising. In *Proceedings of the 9th workshop on Mobile computing systems and applications*, HotMobile '08, pages 80–85, New York, NY, USA, 2008. ACM.

[22] S. Nath, F. X. Lin, L. Ravindranath, and J. Padhye. Smartads: bringing contextual ads to mobile apps. In *Proceeding of the 11th annual international conference on Mobile systems, applications, and services*, MobiSys '13, pages 111–124, New York, NY, USA, 2013. ACM.

[23] M. Papandrea, S. Giordano, S. Vanini, and P. Cremonese. Proximity marketing solution tailored to user needs. *2012 IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, 0:1–3, 2010.

[24] T. Pedersen. Information content measures of semantic similarity perform better without sense-tagged text. In *NAACL*, HLT, 2010.

[25] R. Barnes. *ShopSavvy*. `http://www.biggu.com`.

[26] A. Ranganathan and R. H. Campbell. Advertising in a pervasive computing environment. In *Proceedings of the 2nd international workshop on Mobile commerce*, WMC '02, pages 10–14, New York, NY, USA, 2002. ACM.

[27] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl. Grouplens: an open architecture for collaborative filtering of netnews. In *CSCW*, pages 175–186, 1994.

[28] P. Resnick and H. R. Varian. Recommender systems. *Communications of the ACM*, 40(3):56–58, Mar. 1997.

[29] M. C. Sala, K. Partridge, L. Jacobson, and J. B. Begole. An exploration into activity-informed physical advertising using pest. In *Proceedings of the 5th international conference on Pervasive computing*, PERVASIVE'07, pages 73–90, Berlin, Heidelberg, 2007. Springer-Verlag.

[30] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl. Itembased collaborative filtering recommendation algorithms. In *WWW*, pages 285–295, 2001.

[31] Shopkick. `http://www.shopkick.com/`.

[32] Siri. `http://www.apple.com/ios/siri/`.

[33] J. Yan, N. Liu, G. Wang, W. Zhang, Y. Jiang, and Z. Chen. How much can behavioral targeting help online advertising? In *Proceedings of the 18th international conference on World wide web*, WWW '09, pages 261–270, New York, NY, USA, 2009. ACM.

[34] K. Zhang and Z. Katona. Contextual advertising. *Marketing Science*, 31(6):980–994, Nov. 2012.