ReviewCollage: A Mobile Interface for Direct Comparison Using Online Reviews

Haojian Jin[1,2], Tetsuya Sakai[1,3], and Koji Yatani[1]

[1]Microsoft Research Asia, Beijing, China
[2]Yahoo Labs, Sunnyvale, CA, USA
[3]Waseda University, Tokyo, Japan

haojian.jin@yahoo-inc.com, tetsuyasakai@acm.org, koji@microsoft.com

ABSTRACT

Review comments posted in online websites can help the user decide a product to purchase or place to visit. They can also be useful to closely compare a couple of candidate entities. However, the user may have to read different webpages back and forth for comparison, and this is not desirable particularly when she is using a mobile device. We present ReviewCollage, a mobile interface that aggregates information about two reviewed entities in a one-page view. ReviewCollage uses attribute-value pairs, known to be effective for review text summarization, and highlights the similarities and differences between the entities. Our user study confirms that ReviewCollage can support the user to compare two entities and make a decision within a couple of minutes, at least as quickly as existing summarization interfaces. It also reveals that ReviewCollage could be most useful when two entities are very similar.

Author Keywords

Comparison; summarization; user-generated review; mobile interface; natural language processing.

ACM Classification Keywords

H5.2. Information interfaces and presentation (e.g., HCI): User Interfaces.

General Terms

Human Factors

INTRODUCTION

Marketing research shows that people often perform two-stage decision making to determine their choice of purchase or activities [2, 21]: screening and comparing. Screening means that the user attempts to narrow down to several candidates (in most cases, two or three [13]) from available choices. The user would do this based on her major requirements (e.g., the location and opening hours). This is well supported in many existing websites, such as filtering by several pre-defined parameters.

MobileHCI 2014, September 23–26, 2014, Toronto, ON, Canada.

http://dx.doi.org/10.1145/2628363.2628373

However, comparing is often not well supported in such websites. In this process, the user would check details of the candidates from different perspectives and make the final decision. The user has to manually compare multiple entities by looking for information about particular aspects or reading review comments. Our pilot study with eight users (explained later) found that this is not desirable particularly when the user is using a mobile device as the screen size is limited. Nevertheless, comparison in a small screen is becoming a common activity; nowadays the user often reads webpages to find a place to visit (e.g., seeking a restaurant in an unfamiliar city) through mobile services.

Summarization systems can help the user read review comments as they allow her to view information quickly. Review Spotlight [27] and RevMiner [12] are found effective for quick impression formation. Their work also shows that tag-cloud interfaces help to overview review text. However, summarization performed independently for each entity may not help the comparing process because it does not fully consider relative relationships between multiple entities. This motivates us to develop a mobile interface providing a one-page view that allows the user to closely compare two entities using review comments.

Our system, ReviewCollage, highlights key descriptions about two entities using a tag-cloud interface. It uses attribute-value word pairs extracted from the original review

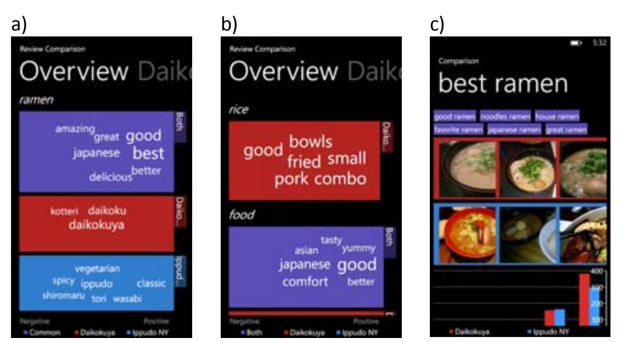

Figure 1. The ReviewCollage interface. a) It shows an overview of two entities (restaurants) by attribute ("ramen" in this view). The three panels show descriptions used for both restaurants (purple) or primarily for one (red and blue). This offers a quick view of how similarly and differently two entities are reviewed; b) The system also shows attributes primarily used for one restaurant; c) The interface also shows details for a clicked word with pictures, rating distributions, and comments.

text, similar to RevMiner [12], to offer an overview of two reviewed entities. Figure 1 shows the ReviewCollage interface with two Japanese restaurants. The three panels show descriptions (values) associated with the attribute "ramen (a soup noodle)". The purple panel contains descriptions representing some features common in the two restaurants. As in Figure 1a, both restaurants are positively described with the terms "best" and "good". The red and blue panels show descriptions used primarily for only one of the restaurants. For example, Daikokuya (in the red panel) is described with "kotteri (thick soup)" whereas the ramen in Ippudo (in the blue panel) is "spicy". In this manner, the user can view how similarly or differently two reviewed entities are described within a one-page view. ReviewCollage also offers the detailed information about a specific word pair (Figure 1c). This view includes pictures, rating distributions, and comments containing the selected word pair, and helps the user deepen their understanding of the entities.

Our comparative user study with existing summarization interfaces using word pairs (Review Spotlight [27] and RevMiner [12]) shows that participants were able to view information about two entities and make a choice at 105 seconds on average. This was significantly faster than RevMiner and comparably fast to Review Spotlight. It also reveals that ReviewCollage could be most useful when two reviewed entities are very similar and summarization for one entity does not highlight differences well.

The contributions of this work are summarized as follows:

- The design of the mobile interface to support a comparison process using review comments;
- The development of Shared-Distinct value, an algorithm to identify attribute-value word pairs used commonly in two entities (shared terms) or primarily for one entity (distinct terms); and
- Our evaluation to confirm the utility of ReviewCollage and highlight different benefits of shared and distinct terms for supporting comparison tasks.

RELATED WORK

Review summarization has been investigated extensively in the field of computational linguistics [11], and user interfaces also have been examined. Liu et al. built a system to visualize how many positive or negative reviews were posted about features of an entity using bar graphs [18]. Carenini et al. [3] used a Treemap visualization to show a summary of reviews although their user study found that participants often preferred text-based summaries because the interpretation of Treemap was often not intuitive.

One common technique used in review summarization is keyword/tag extraction. These keywords and tags allow the user to view information about an entity at a glance. They are often associated with sentiment, in particular, positivity and negativity. Dave et al. [6] built a system to extract tags from product reviews with a sentiment score. Lee et al. [16] developed a system in which the user can manually add tags and their sentiment to an entity, and the rated valence of the tag is shown by the font. The system by Ganesan et al. [9] uses a tag cloud interface with emoticons to visualize eBay seller feedback (e.g., automatically adding a smiley face to a positive tag). Yatani et al. developed Review Spotlight [27] using adjective-noun word pairs in a tag cloud interface. It offers more detailed information at a glance than single-word tag clouds. Their study revealed that Review Spotlight could help the user quickly develop an impression about a reviewed entity (a restaurant in their prototype).

As the screen size is limited on mobile devices, review summarization would become even more beneficial. The study by von Reischach et al. [26] confirms that people prefer ratings and text summaries because they can glance at review comments on a mobile device. Have2eat [8] is a mobile application that offers information about nearby restaurants. It shows its summary about a restaurant by automatically selecting salient sentences in the review text. RevMiner [12] summarizes review text using attribute-value pairs in a manner similar to Review Spotlight [27] to offer a concise overview of how a restaurant is described. But these systems only provide information about one entity, and are not designed to directly support the comparison process.

Prior work has also developed interactive systems that aim to support the comparison process. For example, systems developed by Carenini et al. [4] and Zhang et al. [27] show the distribution of positive and negative reviews or differences about multiple entities along different aspects. ManyLists [19] provides a table view of food nutritional ingredients or product features. It utilizes the spatial layout and animation to indicate similarities between entities as well as their unique features.

These systems categorize information based on pre-determined aspects, and may not highlight the similarities and differences between multiple entities in a flexible and lower-level manner. ReviewCollage addresses this issue by parsing review text and finding descriptions used commonly in two entities and primarily for one of them. Our interface is also designed towards mobile devices, thereby differentiating our work from the projects above.

PILOT STUDY

We conducted a pilot study with eight participants (four male and four female) to inform the interface design for review comment comparison. We chose four sets of two or three restaurant review webpages in the same category. The participants were asked to find out how much they would like to visit each of the restaurants assuming that basic conditions are equal (e.g., distance to the restaurant or budget limit). This design allowed us to focus on examining user behavior on comparison rather than searching or screening. They read the two sets of webpages on a laptop and the other two on a mobile touch-screen device. This experimental design was included to uncover interaction difficulty related to the form-factor of a mobile device.

This pilot study revealed the following behavior which our participants showed often.

Frequent page switching: Participants found that switching multiple webpages needed effort, particularly on a mobile device. Also, this often made it hard for the participants to keep track of which restaurant is good for what, as one participant commented:

"It's quite often that I need to switch back to the former page to check the details since I don't want to remember all the things."

This suggests that the interface should offer a one-page overview of multiple entities.

Looking for similarities and differences: We also found that participants sought how similarly or differently people described the restaurants. For example, one participant provided the reason for her choice by articulating the similarities and differences in the two restaurants:

"Both two restaurants seem good and also provide similar foods, but some people mentioned the service is intolerable in the first restaurant, and I didn't find similar comments on the other. That's why I chose the second one."

This implies that the interface should highlight descriptions which apply to multiple entities or to only one of them.

Comparing visually using pictures: We observed that participants often viewed and compared pictures posted by the reviewers in addition to reading comments. These pictures offered a better understanding of the comments or enhanced their impression about a particular restaurant. Thus, pictures should be incorporated into the interface along with relevant information (e.g., keywords or descriptions which can be associated with them).

These findings motivated us to consider a summarization interface that covers the similarities and differences between two reviewed entities, to facilitate low-level comparison in an efficient manner. Summarization using word pairs and tag-cloud interfaces [12, 27] is found to be effective for quick impression formation. This can also be appropriate for mobile interfaces as it does not require a large visual space. We thus decided to investigate how a tag-cloud interface with word pairs can be extended to support comparison tasks. This requires two improvements over existing systems:

- Determining whether a word pair is frequently used in both entities or primarily in one of them,
- Displaying word pairs to highlight similarities and differences between two entities.

In the following sections, we explain our implementation of ReviewCollage by describing each of the major system components: word pair extraction, word pair ranking, and mobile interface.

WORD PAIR EXTRACTION

We carefully re-implemented the RevMiner extraction method [12]. We first generated attribute-value pairs as initial seeds. We parsed sentences in review text for 20 randomly-chosen restaurants with the Stanford parser [5], and extracted sets of nouns and their modifying adjective. We chose the 50 most frequently used nouns, and the three most frequently used adjectives for each noun, resulting in 150 pairs (the nouns as attributes and the adjectives as values) as the initial seeds. We then performed the bootstrapping algorithm (the details are available in [12]) to collect new attribute-value pairs. The cutoff threshold for our dataset was experimentally set to 11. This needs to be calibrated for other datasets as discussed in [12].

During development, we also came across the same issue of word pair over-generation as RevMiner, such as "happy guys" from the sentence "the guys were happy with the food". These spurious extractions were caused by short templates like "[attribute] were [value]" because it could aggressively match to sentences. To further restrict these short templates, our system also includes templates that consider the words preceding or following the pairs.

We crawled reviews for 48 restaurants in Yelp.com. This resulted in 780k sentences. After performing word-pair extraction, we had 556 templates, 1478 attributes, and 2172 values in our current prototype.

WORD PAIR RANKING

After the extraction, the system attempts to identify which word pairs are used commonly in both restaurants (referred to as "shared terms") or mostly for one restaurant (referred to as "distinct terms"). Monroe et al. [20] examined word classification methods with political content to uncover what word was used primarily by Democrats or Republicans in the United States of America. They tested various approaches, including model-based ones. These approaches could be applicable, but require explicit assumptions for models. These models may need re-training or modifications when they apply for another corpus. We thus sought model-free methods that satisfy the following requirements:

Req1. Our algorithm must differentiate shared and distinct terms only using the occurrence probabilities in two review text sets.

Req2. Our algorithm must weigh shared and distinct terms based on the occurrence probabilities in two review text sets. Suppose that one word pair has a 50% occurrence probability in both review sets, and another has 10%. The former should have a larger weight than the latter.

TF-IDF [24] is commonly used to extract words that occur frequently in particular documents but rarely in the rest of the entire document set. We can apply TF-IDF to our case by using its maximum value in the two review text for example:

$$TF\text{-}IDF(tf_A, tf_B) = Max(tf_A, tf_B) \times log\frac{2}{dc(tf_A, tf_B)},$$

where

$$dc(tf_A, tf_B) = \begin{cases} 2 \ (tf_A \neq 0 \text{ and } tf_B \neq 0) \\ 1 \ (\text{otherwise}) \end{cases},$$

and tf_A, tf_B are term frequencies (occurrence probabilities of a term) in the review text for entity A and B, respectively. A term frequency is defined as the number of occurrences of a term over the number of occurrences of all terms of interest. We do not consider the case of $tf_A = tf_B = 0$ as it means that the term does not appear in either of the review text. This TF-IDF is not generally useful for ReviewCollage. For example, TF-IDF can be zero even in a case where the term is used very frequently in one entity, and very rarely, but not at all, in the other. We want to classify such a term as distinct.

To this end, we explored different existing algorithms commonly used for representing similarities or differences, and also developed our own. We present them with mathematical definitions in this section.

Binary Entropy

Binary entropy [25] represents the average uncertainty in a random variable which can take only two values. In our context, this can be regarded as how likely a word pair would be seen in one entity over the other. If a term is distinct, it is more likely to be observed in one entity; thus, an event of word pair occurrences is certain (or a prediction of which entity has this word pair is certain). For shared terms, the event becomes more uncertain. Thus, Binary entropy becomes small for distinct terms and large for shared terms. Therefore, it can be used to represent similarities on phenomena described by two probability values. The binary entropy is defined as follows:

$$BE(A,B) = -\frac{tf_A}{tf_{A+B}} \log_2\left(\frac{tf_A}{tf_{A+B}}\right) - \frac{tf_B}{tf_{A+B}} \log_2\left(\frac{tf_B}{tf_{A+B}}\right),$$

where tf_{A+B} is the sum of the two term frequencies. To make the polarity consistent with the other algorithms we explain below (large values for distinct terms and small for shared terms), we use the inverse binary entropy (IBE):

$$IBE(tf_A, tf_B) = 1 - BE(tf_A, tf_B).$$

IBE takes a value between 0 and 1 by definition; thus, it is already normalized.

Kullback-Leibler Divergence

Kullback-Leibler Divergence (KL-divergence) [14] is used to quantify the difference between two discrete probability distributions. It represents the information lost which happens when one probability distribution is used to approximate the other. It is defined as the difference between cross entropy and self-entropy:

$$D_{KL}(tf_A||tf_B) = -tf_A\left(\log_2(tf_B) - \log_2(tf_A)\right).$$

Note that KL-divergence is not symmetric; generally, $D_{KL}(tf_A||tf_B) \neq D_{KL}(tf_B||tf_A)$. This is not ideal because our ranking method would otherwise require to calculate and calibrate two directionalities. We thus did not consider KL-divergence for further analysis.

Jensen-Shannon Divergence

Jensen-Shannon Divergence (JS-divergence; JSd) [17] solves the non-symmetric behavior of KL-divergence. It represents the similarity of two probability distributions as the total divergence to the average distribution:

$$D_{JS}(tf_A, tf_B) = \frac{1}{2} D_{KL}(tf_A||tf_M) + \frac{1}{2} D_{KL}(tf_B||tf_B),$$

where tf_M is the mean of the two term frequencies ($tf_M = (tf_A + tf_B)/2$). As D_{JS} takes a value between 0 and 0.5 given that tf_A and tf_B are between 0 and 1, it can be normalized: $D_{JS\_n}(tf_A, tf_B) = 2D_{JS}(tf_A, tf_B)$.

Limitations of the algorithms above

The three algorithms above all satisfy Req1. For example, KL-divergence and JS-divergence are zero when $tf_A = tf_B$, and the values become large when $tf_A \ll tf_B$ or $tf_A \gg tf_B$. But these three algorithms do not satisfy Req2 well, in particular for shared terms (e.g., $D_{JS\_n}(0.5, 0.5) = D_{JS\_n}(0.1, 0.1) = 0$). This problem is also observed in other existing methods we surveyed, such as WordScores [15], frequently used in political science. We chose inverse binary entropy and JS-divergence as examples of existing algorithms mainly because they are commonly used in information theory and still clarify the limitation.

Nevertheless, we found that JS-divergence is close to ideal in a case of distinct terms for the two following reasons. First, JS-divergence is symmetric. Second, JS-divergence gives weights for distinct terms as we desire even when either tf_A or tf_B is equal or close to zero. For example, $D_{JS\_n}(0.5, 0) = 0.5 > D_{JS\_n}(0.1, 0) = 0.1$, but $IBE(0.5, 0) = IBE(0.1, 0) = 1$.

Shared-Distinct value

We developed Shared-Distinct value (SD-value; SDv) to determine whether an adjective is used frequently for both entities or only for one of them. The definition is as follows:

$$SDv(tf_A, tf_B) = Max(tf_A, tf_B) \times \left(\frac{\pi}{8} - \tan^{-1}\frac{Min(tf_A, tf_B)}{Max(tf_A, tf_B)}\right).$$

The first term of the product (the maximum of the occurrence ratios in the review text of two entities) reflects Req2. The second term maps shared and distinct terms into positive and negative values, respectively; thus, it satisfies Req1. It also contributes a weight based on the skewness of the ratios. For example, if the ratios are equal, this part becomes the minimum negative value of $-\pi/8$. But if the ratios are skewed in one way, it gives a large positive value towards $\pi/8$ because the arctangent part is close to zero. In this manner, the SD-value can offer both the shared/distinct polarity and magnitude in a single measure. The deduction from $\pi/8$ also ensures that Req2 is met.

Given that A and B are between 0 and 1, the SD-value can take a value between $-\pi/8$ and $\pi/8$. To maintain the sign, we can normalize SD-value into [-1, 1], defined as:

$$SDv_n(tf_A, tf_B) = \frac{SDv(tf_A, tf_B)}{\pi/8}.$$

Our SD-value is only based on word occurrence frequencies and can be used in different corpora without modifications.

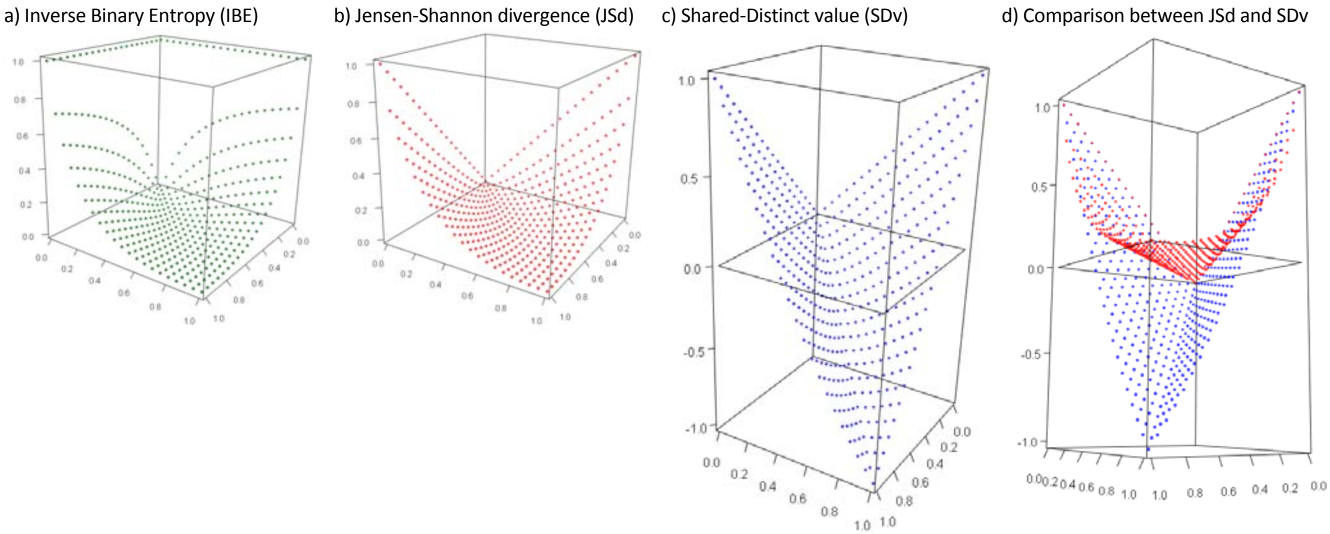

Figure 2. Value distributions of the inverse binary entropy (green), normalized Jensen-Shannon divergence (red), and normalized Shared-Distinct value (blue). In all plots, the horizontal axes represent the occurrence probabilities of a term within each review text of two entities (*A* and *B*). Please refer to the accompanying video to see these plots from different perspectives.

Systematic Examination on Shared-Distinct Value

As we discussed in the previous section, our algorithm must satisfy two major requirements (Req1 and Req2). In addition, it ideally should have a similar characteristic to JS-divergence for distinct terms as it is a commonly-used similarity measurement for two probability distributions.

We conducted simulations to systematically evaluate our SD-value. Figure 2 shows the value distributions of the normalized IBE, JS-divergence, and SD-value by ranging values of tf_A and tf_B from 0 to 1 with an interval of 0.05 (excluding the case of $tf_A = tf_B = 0$). The smoothing factor of 1E-100 was included to avoid the log of zero and division by zero when necessary. The plots of IBE and JS-divergence again illustrate limitations discussed in the previous section. The plot of SD-value shows that shared terms would have negative values, forming the main difference from the other two algorithms. Figure 2d shows plots of JS-divergence and SD-value integrated in one space. As we desired, the distribution of the SD-value closely overlaps with that of JS-divergence for distinct terms. A Pearson's correlation was found highly strong (r^2=.88, p<.001). This also supports that SD-value behaves similarly to JS-divergence for distinct terms and differentiates shared terms based on their occurrence probabilities.

Table 1 shows an example comparison between SD-value and the other algorithms. SD-value can determine whether each word pair is shared or distinct by its sign and absolute value. In the other algorithms, some shared terms are not weighed properly (e.g., TF-IDF and IBE), or distinct terms are likely to be mis-classified as shared (e.g., "awesome food" with JS-divergence).

Based on the evidence gained through our quantitative examination, we decided to use SD-value in ReviewCollage. After the word pair extraction, ReviewCollage first ranks all the attributes with the absolute SD-value. It then distinguishes all associated values for each attribute into shared and distinct word pairs using the signed SD-value, and ranks each of them. This ranking result is used to prioritize word pairs to be shown in the interface, as explained in the next section.

Word pair	Term frequency		Algorithms (normalized)			
	A	B	TF-IDF	IBE	JSd	SDv
Good food	0.1013	0.0767	0	0.0138	0.0025	-0.0659
Great food	0.0452	0.0625	0	0.0187	0.0020	-0.0372
Tasty food	0.0095	0.0130	0	0.0175	0.0004	-0.0079
Bad food	0.0047	0.0047	0	0.0000	0.0000	-0.0047
Indian food	0.5948	0	1	1	0.5948	0.5948
Flavorful food	0.0060	0	0.0100	1	0.0060	0.0060
Spicy food	0.0202	0.0035	0	0.3960	0.0094	0.0114
Chinese food	0.0023	0.6033	0	0.9640	0.5838	0.5974
Awesome food	0.0024	0.0106	0	0.3099	0.0040	0.0046

Table 1. An algorithm comparison with word pairs of "food" extracted from two actual restaurant reviews. The values are normalized within each algorithm. Our SD-value can separate shared and distinct terms by its sign and absolute value.

REVIEWCOLLAGE INTERFACE DESIGN

After identifying shared and distinct word pairs, ReviewCollage displays them in an interface using tag cloud visualization. ReviewCollage has two views to support the user's comparison: Summary View and Detail View. A complete demonstration of the ReviewCollage interface is available in the accompanying video.

Summary View

When the user chooses two entities, ReviewCollage shows panels of tag clouds in a one-page view (Figure 1a). We use a tag cloud interface because it can visually emphasize representative words [12, 22, 27]. The three panels contain descriptions (values of the extracted word pairs) associated with a particular attribute ("ramen" in Figure 1a). The system does not use high-level or generic concepts; thus, the interface offers comparison on low-level aspects (e.g., a specific type of dishes), and would help the user compare two entities closely. ReviewCollage also shows attributes that appear mainly in one restaurant to highlight their unique aspects as shown in Figure 1b.

The three panels under each attribute show associated values used to describe both entities (purple: Shared panel) and those mostly used for one restaurant (red and blue: Distinct panel). The user can customize the color scheme if desired. Each panel also has a label: "Both" for the shared word pairs and the entity's name for the distinct word pairs. This view thus highlights similarities and differences between the two entities. For example, the interface shows that ramen (soup noodle) in both entities are positively described because "best", "good", and "great" are in the purple panel. But the ramen in Daikokuya (red) can be characterized as "kotteri (thick soup)" whereas the one in Ippudo (blue) is described as "spicy" or "vegetarian".

ReviewCollage uses the absolute SD-value to order the attributes extracted from the review text, starting with the largest to the smallest except for the ones under the threshold. The font size of values for each attribute is set to be proportional to their absolute SD-value to highlight salient words. The system also performs sentiment analysis using SENTIWORDNET [7]. It uses the average sentiment value across all the senses (meanings) for each value. This is used to determine the layout of the adjectives in the panel. Our layout algorithm attempts to loosely place positive and negative adjectives towards right and left sides, respectively. In this manner, the user can glance how positively or negatively entities are mentioned from the word layout.

Detail View

When the user taps a word in a panel, the system provides the details about the selected word pair (Figure 1c). Precise pointing with a finger is difficult, and ReviewCollage employs object-based target selection. When the user makes a tap within a panel, the system calculates the distance from the centroid of each word, and determines the one closest to the contact point under the threshold (4 mm) as the target word. Detail View first shows pictures posted in the original review webpage and associated with the chosen word pair. We found that the interface shows only few pictures for some word pairs because descriptions associated with most of the pictures do not match with them. Thus, we extract pictures associated with the attribute-value word pair as well as with the attribute only, and prioritize the ones associated with the word pair in the view. Our informal study confirmed that participants would prefer to see many pictures even if some do not exactly match to the selected word pair.

The view shows the rating distributions of the reviews associated with the selected word pair. They are based on the overall ratings in the original webpage. Thus, this graph does not offer the exact sentiment distribution about the word pair; however, the graph still provides a quick view of how positively or negatively people comment on the entities when they mention the selected word pair.

The system shows several sentences containing the selected word pair. Each sentence is again associated with the overall rating, and the system attempts to match the rating distribution of the sentences to the one shown in the graph. We also prioritize sentences whose length is between 10 and 30 words. This criterion was chosen based on heuristics that very short sentences would not offer much detail while very long sentences would take too much time to read. The system then orders the candidate sentences by the "usefulness" score (the number of votes given by users who found the review useful in the original webpages), and shows most highly ranked sentences for each rating.

Examples Using Review Text in Other Domains

We have so far explained the interface design and features of ReviewCollage with reviews on restaurants (extracted from Yelp.com), but it can be extended to other types of review text. Figure 3 shows Summary View examples using Amazon and TripAdvisor reviews. Similar to the example shown in Figure 1, the user can see similarities and differences between two entities. For example, in Figure 3b, service in two hotels of interest receive both positive and negative descriptions. But in the panels for distinct terms, one of them shows strongly positive words, such as "amazing" and "outstanding". This indicates that service in that hotel may be substantially better than the other in general.

This demonstration suggests that ReviewCollage generally satisfies our main requirement: highlighting similarities and differences between two reviewed entities from different domains (restaurants, hotels and products). An extensive evaluation on how different domains could influence the word pair extraction and interface is beyond the scope of this work. Instead, we evaluated ReviewCollage through a user study to compare it with other summarization interfaces.

USER STUDY

To validate our design of ReviewCollage, we conducted a comparative study against existing summarization user interfaces using word pairs: Review Spotlight [27] and RevMiner [12]. We opted out original webpages because the experiment otherwise would bias towards ReviewCollage. We decided to include RevMiner as it is the latest mobile summarization interface. Although Review Spotlight was not designed for mobile devices, we decided to include it because ReviewCollage takes a similar approach to summarization. The study on RevMiner [12] also shows some advantages of a mobile interface like Review Spotlight. We re-implemented Review Spotlight and RevMiner as faithfully as we could for our study, as shown in Figure 4.

Experimental Design

We crawled review comments on 16 restaurants (8 restaurant pairs) from Yelp.com. Two of these restaurant pairs were in the same category (American, Barbeque, Chinese, and Japanese). We chose these categories because they

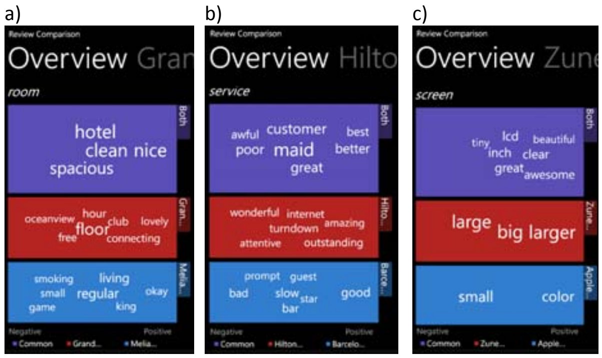

Figure 3. ReviewCollage with reviews in different domains: a, b) hotel reviews ("room" and "service attributes) from TripAdvisor; and c) product reviews (mp3 players) from Amazon.

Figure 4. Three summarization interfaces used in our user study. For Review Spotlight and RevMiner, two views were prepared and each showed one restaurant.

Restaurant Category	Group A			Group B		
	RC	RS	RM	RC	RS	RM
American	C	C	d	C	d	C
Barbeque	C	d	C	C	C	d
Chinese	C	C	d	C	d	C
Japanese	C	d	C	C	C	d

RC: Review Collage, RS: Review Spotlight, RM: RevMiner
C: Comparison pair / d: Distractor pair

Table 2. Experimental conditions. Participants were divided into Group A or B randomly. The distractor restaurant pair was introduced to prevent participants from directly using their previous decision.

represented different types of food and places. These restaurants also had a sufficient number of reviews at the time of this experiment. One of the two pairs in the same category was used for comparing ReviewCollage with either Review Spotlight or RevMiner (referred as a *co mparison pair*), and the other pair was used as a distractor with the remaining interface (referred as a *distract or pair*). This distractor was introduced to prevent participants from directly using their previous decisions by simply judging the restaurant category. Table 2 shows the two groups randomly assigned to six participant each.

As the Review Spotlight and RevMiner interfaces only show a summary about one restaurant, we prepared two views for each comparison task. Participants could switch between the views by swiping horizontally on the screen. We considered side-by-side views of these reference systems in a landscape mode. But it would require frequent up/down scrolls, and impose another difference (e.g., portrait vs. landscape views) to this experiment. Our re-implementation does require a horizontal swipe, but this is a very quick, transient gesture. The view also offers larger space to show information for each entity. We thus determined that the implementation was acceptable for our purpose. Detail View was available for any value word in ReviewCollage and RevMiner or word pair in Review Spotlight. The names of the restaurants were anonymized in all interfaces.

The presentation order of the twelve tasks (3 interfaces × 4 categories) for each participant was randomized, but the presentation of each comparison pair was separated by at least two tasks. We counter-balanced the presentation order of the interfaces for each comparison pair across participants. As our comparison focuses on the interfaces, we did not analyze the performance across restaurant categories.

Procedure

At the beginning of the experiment, participants were given instructions of the study and system, and time to practice with the interfaces until they felt comfortable with using them. After participants chose one restaurant in each task, the experimenter asked them to describe their reasons. They were instructed to tell a couple of points which made them choose that particular restaurant. For task completion time, we measured the time between when an interface was shown to the participants and when they stopped using it and described their decision to the experimenter.

After the participants completed the twelve tasks, they were invited to a short interview to express their opinions on the three interfaces. We did not directly ask preferences to avoid potential good-subject effect. Rather, we asked them to express what benefits and shortcomings each interface have and how they would use it in a realistic setting. The entire study took approximately one hour. A gift card worth $15 USD was offered as the compensation.

Participant

Twelve participants (P01–P12; seven male and five female; average age 23.3 years old) from our research institute were recruited for this study. They were undergraduate or graduate students with various backgrounds in different universities. Some of the participants were non-native English speakers though all of them were able to understand the instructions and express their opinions in English. All had experience of using mobile touch-screen devices, and were aware of or had used mobile apps showing restaurant information and reviews. None of them participated in our pilot study.

RESULTS

Task Completion Time

Figure 5 shows the average task completion time with the three interfaces. Mauchly's test did not show a violation of sphericity (an assumption of a repeated-measure ANOVA test). This permits direct interpretation of one-way repeated-

Figure 5. Task completion time. The error bar represents the 95% confidence interval.

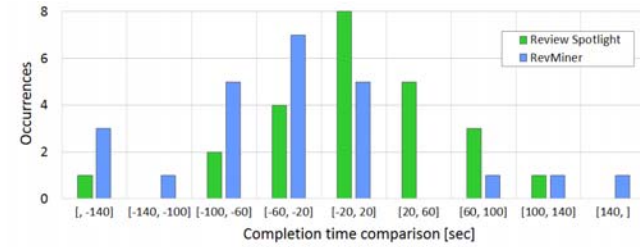

Figure 6. The distribution of the task completion time differences between ReviewCollage and the other interfaces. A negative value represents a case in which participants finished the task more quickly with ReviewCollage than with the other interface.

measure ANOVA F-test results. An ANOVA test on the completion time against the interface with all tasks showed a significant difference on the completion time ($F_{(2, 22)}$=5.52, p<.05, η_p^2=.33). The post-hoc pairwise comparison with paired t-tests using the Bonferroni correction reveals a significant difference, showing that Review Spotlight were faster than RevMiner (p<.01).

An ANOVA test with all the tasks excluding the distractors also found a significant effect of the interfaces on the completion time ($F_{(2, 22)}$=6.13, p<.05, η_p^2=.36). The post-hoc test reveals significant differences with ReviewCollage and Review Spotlight outperforming RevMiner (p<.05 for both). Figure 6 shows the distribution of completion time differences between ReviewCollage and the other interfaces. This plot confirms that ReviewCollage was comparably fast to Review Spotlight and generally faster than RevMiner.

Effect of Interfaces on Decision-making

The participants made the same restaurant choice for 34 out of the 48 tasks with the comparison pairs (70.8%). We found that the same participants often gave different reasons for their choices even when they chose the same restaurant. For example, one participant was able to express a specific reason on her choice with ReviewCollage:

"I gonna choose Restaurant A. Because I didn't like spicy, salty, hot things. And Restaurant A has Cantonese food, and I haven't tried any Taiwanese food before. I don't want to try something new this time." [P02, ReviewCollage, 135 sec (completion time)]

On the other hand, this participant was struggling to make a decision on the same comparison pair with RevMiner. As a result, she had to spend more time, but was not able to express specific reasons.

"I would say, Restaurant A. It's really hard to choose. Both restaurants' price is reasonable. But Restaurant A has long waiting, and Restaurant B is also the same, long wait. I don't see any obvious thing here." [P02, RevMiner, 201sec]

Some participants often made their decisions with RevMiner based on frequency numbers shown next to words by assuming that higher numbers indicate that more people had visited the place.

Participants used a few word pairs as part of reasons for their choices with RevewCollage. This was also true when they used Review Spotlight, but they tended to mention the differences rather than the similarities with ReviewCollage:

"I choose Restaurant B because there is a comment like dishes in Restaurant A are dirty or something. And it is about Taiwanese food, and I want to have a try." [P05, ReviewCollage, 102 sec]

"I choose Restaurant B because comments mention more different dishes. And there are a lot of things in common, but comments in Restaurant B have 'cup chicken', which may be a famous Chinese dish. I am more interested in it." [P05, Review Spotlight, 156 sec]

We observed a similar reasoning pattern when participants made different choices. ReviewCollage often helped participants quickly find aspects which they were interested in but was not visible in the other interfaces, and this led to changes in their choices. For instance, P09 described the reasons for his choices with ReviewCollage and Review Spotlight as follows:

"I would choose Restaurant B. Because I am so much into spicy food, and I saw keywords with 'spicy' in Restaurant B, but not in A." [P09, ReviewCollage, 26 sec]

"I chose Restaurant A. I see both of them have Chinese food. I checked the details of them, and thought A is better becau se I saw more positive comments." [P09, Review Spotlight, 89 sec]

DISCUSSIONS

Our results show that ReviewCollage was able to support comparisons and informed decisions at least as fast as existing interfaces using word pairs. We further examine differences between ReviewCollage and the other interfaces through comments collected in post-experiment interviews.

Uncovering Different Use Scenarios

Our quantitative results show that ReviewCollage and Review Spotlight performed comparably in terms of completion time. One reason is that both interfaces are based on tag clouds, and they were easy to glance. This is in line with findings from prior work [12, 22, 27]. More specifically, participants expressed that Review Spotlight could offer an overview that was quicker to read whereas ReviewCollage could show the details better.

"[ReviewCollage] separates common keywords for both restaurants, so I can focus on something special for each restaurant. But if I want to make a choice quickly, I would use [Review Spotlight] because it is easier to see biggest keywords. But if I want to see more details about two restaurants, I would use [ReviewCollage]." [P03]

Another difference between ReviewCollage and Review Spotlight was that word pairs are grouped by attribute in ReviewCollage, and participants were able to quickly focus on specific aspects of restaurants they were interested in:

"It has some good things. I especially like noodles. I want to eat noodles in these restaurants. So I would like to compare feedback on each restaurant about noodles. And [ReviewCollage] has specific categories, and that is useful because it can separate noodles from sauce and soup, for example." [P02]

These differences indicate that ReviewCollage and Review Spotlight have different benefits despite the similarity in the interface design. In particular, participants considered ReviewCollage to be more useful when the two entities to be compared were very similar as one participant commented:

"To compare two restaurants, I thought this interface (ReviewCollage) would be much better especially when they are the same types of restaurants. For example, if I choose the same type of restaurants and I want to compare, I want to see what are common things and different things. If there is a difference, for example, a Chinese restaurant and Japanese restaurant, I guess, another one (Review Spotlight) would be better. You would just see keywords for each restaurant, and that would be enough." [P12]

When two entities are relatively dissimilar, Review Spotlight can show many different word pairs between them, and thus can highlight unique aspects of each entity. However, when they are very much alike (e.g., restaurants in the same food category), Review Spotlight would mostly display similar word pairs. This requires the user to visually search different word pairs in two views. ReviewCollage can emphasize the differences between two entities even for such a case.

Advantages of Tag Clouds

Our user study found that ReviewCollage was significantly faster than RevMiner. One reason was that RevMiner does not visually highlight frequently-used word pairs well as compared to ReviewCollage, and participants tended to spend more time finding word pairs they thought were important or representative of restaurants.

"I couldn't clearly see which words were more important because the size of words was the same." [P08]

Participants also found that they often needed to spend much time to identify the differences between two entities in RevMiner, as they did with Review Spotlight, because shared attributes and values were not grouped well.

"I was a little confused with labels. I thought there were the same labels (attributes) in both screens, so I was trying to find the same label for comparison. But it was not always the case. So it was often hard to compare." [P09]

Participants liked the number next to each word showing its occurrence frequency. Some participants extrapolated that higher numbers indicate that more people had visited the place, and used it as part of reasons. But, they generally felt that RevMiner is not organized for comparison, and this was also reflected in the performance time.

Different Benefits of Shared and Distinct Terms

ReviewCollage shows word pairs that represent both restaurants (shared word pairs) and ones that apply to only each of the restaurants (distinct word pairs). Our interview revealed that many of our participants used Distinct panels more often than Shared panels.

"I found Both (shared word pairs) is not helpful for me to judge because I only needed to see the difference between Restaurant A and B. Maybe it is helpful for me to judge whether I want to go to these two restaurants." [P10]

However, some participants also found shared panels useful. As we expected, they used Shared panels to ensure that some aspects important to them are equally positive or negative. In addition, they used Detail View for words in Shared panels to perform comparison as P08 commented:

"[ReviewCollage] is useful for comparison, especially when I tapped into a word (in Shared panels) and it gave me pictures and a chart about both restaurants." [P08]

These findings show that ReviewCollage can support different ways to perform comparisons, and validate its interface design of showing the similarities and differences between two reviewed entities in a one-page view.

LIMITATIONS

Our present study has several limitations. First, the user study design did not investigate how satisfied participants would be about decisions they made in light of experience at an actual restaurant. We selected restaurants located in a different city to ensure that the participants would not be able to use prior experience with them. A deployment study would help to examine user satisfaction on decision making through ReviewCollage in a realistic setting.

The sample size is relative small (twelve participants). This could be one reason why we did not see clear quantitative differences among the conditions. However, our qualitative results indicate important differences, and encourage future work to examine them deeply in a larger scale.

Our summarization is not always perfect like other systems. Some heuristics may help, but a generalizable approach would be challenging. This is an active research topic in computational linguistics, and future improvements could be integrated into the ReviewCollage system. We do not claim that SD-value is the optimal solution, and future work should examine various ranking algorithms systematically. This work offers a foundation for such future studies.

CONCLUSION AND FUTURE WORK

Comparison between multiple reviewed entities on a mobile device is becoming a common activity. But current interactive systems do not support this process well because existing summarization methods only take one entity into account. We present ReviewCollage, a mobile interface to support direct comparison between two entities using review comments. ReviewCollage extracts attribute-value word pairs from reviews, and highlights descriptions that apply to both entities and to only one of them. Our user study confirms that ReviewCollage can help comparison and decision-making within a couple of minutes. This was at least as fast as existing summarization interfaces using word pairs. It also reveals that ReviewCollage can be most useful when two entities are very similar.

Future work should test different visualization methods. Word Spectrum [10] could be one approach to showing shared/distinct word pairs in one panel although it requires large landscape view space. Adaptively changing the visualization by the device orientation would be one solution.

We plan to investigate how ReviewCollage can be extended to support the comparison among more than two entities. While our SD-value can be readily extended to more than two entities (e.g., by using an Euler angle for three entities), the interface showing differences among more entities can become very complex, and is a challenge for future work.

ACKNOWLEDGEMENTS
We are grateful to Jeff Huang for providing the details of the implementation of the RevMiner attribute-value extraction method. We also thank all our user study participants.

REFERENCES

1. Baccianella, A., Esuli, A., Sebastiani, F. SentiWordNet 3.0: an enhanced lexical resource for sentiment analysis and opinion mining. In *Proc. LREC 2010*, ELRA, 2200-2204.
2. Beach, L. R. Broadening the definition of decision making: The role of prechoice screening of options. *Psychological Science 4*, 1993, 215-220.
3. Carenini, G., Ng, R. T., Pauls, A. Interactive multimedia summaries of evaluative text. In *Proc. of IUI 2006*, ACM, 124-131.
4. Carenini, G., Rizoli, L. A multimedia interface for facilitating comparisons of opinions. In *Proc. of IUI 2009*, ACM, 325-334.
5. Cer, D., Galley, M., Jurafsky, D., Manning, C. D. Phrasal: a toolkit for statistical machine translation with facilities for extraction and incorporation of arbitrary model features. In *Proc. of HLT-DEMO 2010*, ACL, 9-12.
6. Dave, K., Lawrence, S., Pennock, D. M. Mining the peanut gallery: opinion extraction and semantic classification of product reviews. In *Proc. of WWW 2003*, ACM, 519-528.
7. Esuli A., Sebastiani F. SENTIWORDNET: a publicly available lexical resource for opinion mining. In *Proc. of LREC 2006*, ELRA, 417-422.
8. Fabbrizio, G. D., Gupta, N., Besana, S., Mani, P. Have2eat: a restaurant finder with review summarization for mobile phones. In *Demo Volume of COLING 2010*, ACL, 17-20.
9. Ganesan, K. A., Sundaresan, N., Deo, H. Mining tag clouds and emoticons behind community feedback. In *Proc. of WWW 2008*, ACM, 1181-1182.
10. Harrison, C. Word Spectrum: Visualizing Google's Bi-Gram Data. http://www.chrisharrison.net/index.php/Visualizations/WordSpectrum
11. Hu, M., Liu, B. Mining and summarizing customer reviews. In *Proc. KDD 2004*, ACM, 168-177.
12. Huang, J., Etzioni, O., Zettlemoyer, L., Clark, K., Lee, C. RevMiner: an extractive interface for navigating reviews on a smartphone. In *Proc. of UIST 2012*, ACM, 3-12.
13. Hubl, G., Trifts. V. Consumer decision making in online shopping environments: the effects of interactive decision aids. *Marketing Science 19* (1), 2000, 4-21.
14. Kullback, S, Leibler, R. A. On information and sufficiency. *The Annals of Mathematical Statistics 22.1*, 1951, 79-86.
15. Laver, M., Benoit, K., Garry, J. Extracting policy positions from political texts using words as data. *American Political Science Review 97*, 02 (2003), 311-331.
16. Lee, S. E., Son, D. K., Han, S. S. Qtag: tagging as a means of rating, opinion-expressing, sharing and visualizing. In *Proc. of SIGDOC 2007*, ACM, 189-195.
17. Lin, J. Divergence measures based on the Shannon entropy. *IEEE Transactions on Information Theory 37.1*, 1991, 145-151.
18. Liu, B., Hu, M., Cheng, J. Opinion observer: analyzing and comparing opinions on the Web. In *Proc. of WWW 2005*, ACM, 342-351.
19. Liu, R., Chao, T., Plaisant, C., Shneiderman, B. ManyLists: product comparison tool using spatial layouts with animated transitions. *University of Maryland Technical Report*, 2012.
20. Monroe, B. L., Colaresi, M. P., Quinn, K. M. Fightin'words: Lexical feature selection and evaluation for identifying the content of political conflict. *Political Analysis 16*, 4 (2008), 372-403.
21. Payne, J.W., Bettman, J.R., Johnson, E.J. *The adaptive decision maker*. Cambridge University Press, 1993.
22. Rivadeneira, A. W., Gruen, D. M., Muller, M. J., Millen, D. R. Getting our head in the clouds: toward evaluation studies of tagclouds. In *Proc. of CHI 2007*, ACM, 995-998.
23. Rrnyi, A. On measures of entropy and information. In *Proc. Berkeley Symposium on Mathematical Statistics and Probability*, University of California Press, 1961, 547-561.
24. Salton, G., McGill, M. J. *Introduction to Modern Information Retrieval.* McGraw-Hill Inc., New York, NY, USA, 1986.
25. Shannon, C.E. A Mathematical Theory of Communication, *Bell System Technical Journal 27*, 1948, 379-423, 623-656.
26. von Reischach, F., Dubach, E., Michahelles, F., Schmidt, A. An evaluation of product review modalities for mobile phones. In *Proc. MobileHCI 2010*, ACM, 199-208.
27. Yatani, K., Novati, M., Trusty, A., Truong, K. N. Review spotlight: a user interface for summarizing user-generated reviews using adjective-noun word pairs. In *Proc. CHI 2011*, ACM, 1541-1550.
28. Zhang, J., Jones, N., Pu. P. A visual interface for critiquing-based recommender systems. In *Proc. EC 2008*, ACM, 230-239.