RevMiner: An Extractive Interface for Navigating Reviews on a Smartphone

Jeff Huang
University of Washington
uist@jeffhuang.com

Oren Etzioni
University of Washington
etzioni@cs.washington.edu

Luke Zettlemoyer
University of Washington
lsz@cs.washington.edu

Kevin Clark
University of Washington
kevinc22@cs.washington.edu

Christian Lee
University of Washington
clee@cs.washington.edu

ABSTRACT

Smartphones are convenient, but their small screens make searching, clicking, and reading awkward. Thus, perusing product reviews on a smartphone is difficult. In response, we introduce REVMINER—a novel smartphone interface that utilizes Natural Language Processing techniques to analyze and navigate reviews. REVMINER was run over 300K Yelp restaurant reviews extracting attribute-value pairs, where attributes represent restaurant attributes such as *sushi* and *service*, and values represent opinions about the attributes such as *fresh* or *fast*. These pairs were aggregated and used to: 1) answer queries such as "cheap Indian food", 2) concisely present information about each restaurant, and 3) identify similar restaurants. Our user studies demonstrate that on a smartphone, participants preferred REVMINER's interface to tag clouds and color bars, and that they preferred REVMINER's results to Yelp's, particularly for conjunctive queries (*e.g.*, "great food and huge portions"). Demonstrations of REVMINER are available at `revminer.com`.

ACM Classification: H.5.2 [Information interfaces and presentation]: User Interfaces – graphical user interfaces.

Keywords: opinion mining; user reviews; information extraction; information retrieval.

INTRODUCTION

People are using smartphones to access a wealth of information on the Web at an unprecedented rate. Yet, the phones' tiny screens make searching for Web pages, and reading them, far less convenient and efficient than on the desktop. To address this challenge, we investigate *extractive interfaces*—user interfaces that utilize attribute-value pairs automatically extracted from unstructured text to summarize the text concisely.

UIST'12, October 7–10, 2012, Cambridge, Massachusetts, USA.

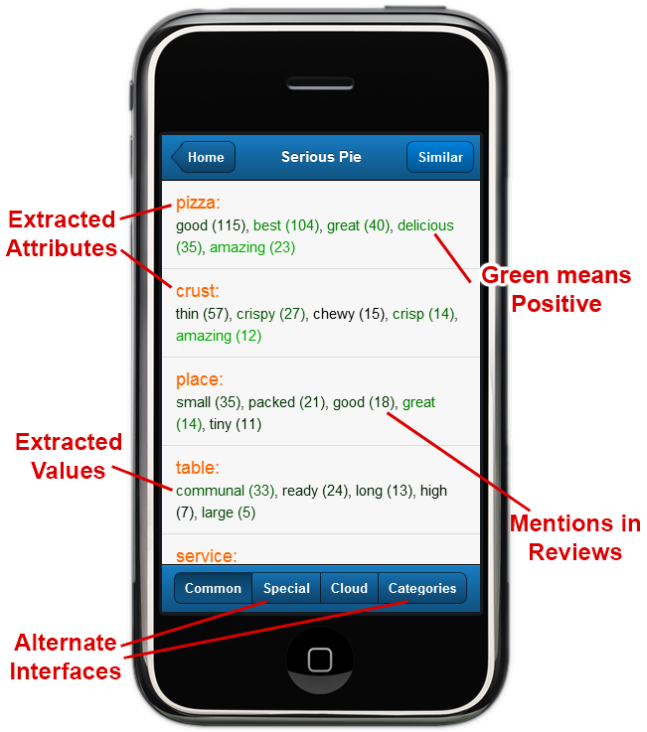

Figure 1: The REVMINER interface listing attributes and associated values extracted from Yelp reviews for the Serious Pie pizza restaurant in Seattle.

This automatically-extracted summary obviates reading the original text in many cases, and enables new approaches to information navigation and retrieval. Instead of keyword search, which often yields a long list of snippets that are awkward to read on the phone, an extractive interface supports querying by a particular attribute-value pair (*e.g.*, "surprising ending" for movies or "high in vitamin C" in the descriptions of fruit and vegetables). Reviews for businesses and products are particularly helpful on mobile devices when a user is ready to act "then and there". Thus, we investigate extractive smartphone interfaces to reviews.

We focused, in particular, on the familiar task of selecting a restaurant based on user reviews. Reviews for businesses and products are particularly appropriate on mobile devices, which are often used when a user is prepared to shop and eat, then and there. We introduce REVMINER, which summarizes reviewer opinions by extracting the most-mentioned attributes (*e.g.*, dim sum, happy hour, service, *etc.*) and associated values (*e.g.*, fresh, outstanding, slow) utilizing state-of-the art Natural Language Processing (NLP) methods. Like earlier work on review mining [12, 13, 21], REVMINER analyzes each value to determine its polarity and strength (*e.g.*, "exquisite" is stronger than "good"). Next, REVMINER assigns a color in the green-red spectrum based on this analysis. Adjacent to each value, REVMINER records the number of reviews in which this particular attribute-value pair appeared (see Figure 1).

Compared with the standard keyword search interfaces, extractive interfaces, like REVMINER's, offer a more structured way to both query and navigate text. Instead of long lists of snippets, REVMINER returns a short list of attributes and their associated values. If more detail is necessary, the user can "click through" to the original text, establishing the provenance of the extraction. In contrast, with interfaces that highlight *phrases* from the text, the structured attribute-value representation enables querying by conjunctions and disjunctions of attributes, and sorting results by the strength of the values.

In summary, REVMINER's compact extractive interface enables smartphone users to conveniently:

query: for restaurants matching one or more attribute-value pairs (*e.g.*, "good margaritas and large portions"),

rank: obtain a list of restaurants sorted on the strength of values for the queried attribute (*e.g.*, a restaurant with "exquisite margaritas" is ranked above one with "good margaritas"),

summarize: view a concise summary of hundreds of reviews organized by attribute and associated value,

identify: restaurants similar to a particular one based on its attributes and peoples' opinions of the attributes

elaborate: access the source of any particular value in its original sentences, and in the underlying reviews.

Finally, the interface also supports querying by restaurant name which brings up a concise summary of the corresponding reviews, culling and aggregating information from a large number of different documents.

In this paper, we ran REVMINER on a set of 300K Yelp reviews about Seattle restaurants; REVMINER has also been run on user reviews covering hotels, beauty salons, and more, demonstrating its generality. In contrast, most previous work on opinion mining was run at far smaller scales. The massive review corpus enables both higher-accuracy extraction and interface elements that rely on statistics computed over the corpus (see the Extraction and Processing section).

Our main contributions are to:

- Articulate and analyze the extractive interface abstraction for smartphones.
- Design and implement the REVMINER extraction system and associated user interface.
- Assess REVMINER's overall performance in a user study that contrasted its interface with the Yelp Android app and a re-implementation of alternate interfaces (*e.g.*, a tag cloud and color bar).

The web site `http://revminer.com` allows anyone to experiment with a version of the interface for the desktop, for the iPhone, and also to download the Android app used in our experiments.

The remainder of this paper is organized as follows. In the next section, we describe the related work and how REVMINER builds on the latest research in review interfaces and NLP. The Extraction and Processing section presents the review data along with how we processed the extractions. The Mobile App section describes the applications of using the extractions to build a full-featured mobile app that allows users to search and be informed about particular restaurants. Finally, we present a user study of two common restaurant tasks where participants evaluate the app. We wrap up with a discussion about going beyond reviews, and conclude.

RELATED WORK

There has been extensive work on opinion mining [12] and even more on Information Extraction (IE) from the Internet [4]. For instance, the OPINE system introduced an unsupervised method for identifying product features and the associated opinions (including their strength and polarity) [13]. While there are technical differences between REVMINER's extraction methods and previous work, our focus in this paper is on the novel *extractive interface* embodied in REVMINER, not on creating a novel extraction algorithm.

While there is an extensive body of work on opinion mining and IE, there has been little work on extractive user interfaces, and none of that work has investigated opinion mining as a substrate for a smartphone interface. One notable exception is the innovative Review Spotlight system, which introduced a tag cloud interface based on adjective-noun pairs extracted from reviews [21].

REVMINER builds on the ideas described by Yatani *et al.* but there are key differences from Review Spotlight. REVMINER utilizes statistical NLP methods for extracting attribute-value pairs, identifies synonyms, and determines the polarity and strength of opinions. These capabilities are enabled by our 300K review dataset which supports probabilistic learning techniques, whereas Review Spotlight generates a tag cloud directly from each review. REVMINER's attribute-value pairs differ from Review Spotlight's adjective-noun pairs. Attributes relate to restaurants, and the values are opinions of the attributes; for example, "only thing" and "other person" are adjective-nouns from the Review Spotlight paper but not attribute-values in REVMINER. Regarding the interface, REVMINER also produces attribute-value summaries in addition to a tag cloud. While Rivadeneira et al. [16] finds that tag clouds are good at impression formation, other in-

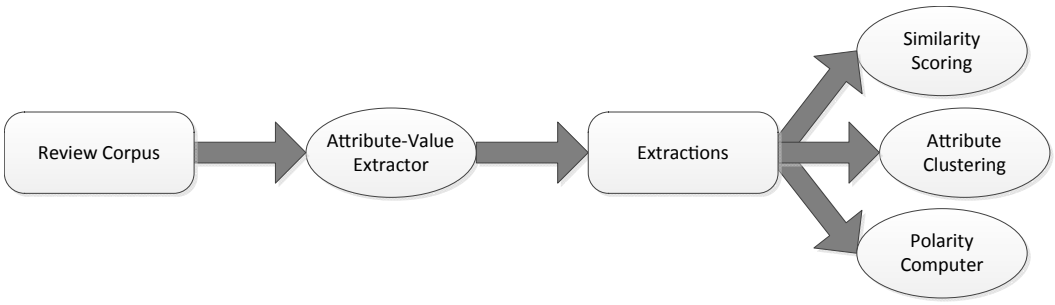

Figure 2: Architecture of the REVMINER offline system. The attributes and values are extracted from the review corpus, which are further processed for other applications. The extraction is performed solely on the review text without any structured data.

terfaces may be better for the browsing and searching tasks that Rivadeneira poses. REVMINER parses search queries to perform a semantic search (*e.g.*, "cheap mexican food free parking" will match places where the *food* is noted to be both *cheap* and *mexican*, at the same time the *parking* is noted to be *free*). Finally, unlike Review Spotlight, our focus is on an extractive interface for a smartphone as evident from our user studies.

Von Reischach et al. evaluate the interaction with product reviews on mobile phones. They remark, "we consider todays product reviews not suitable for mobile phones" [20], referring to review text in their unabbreviated form. Later work by Von Reischach et al. [19] find that text is still rated highest by users for "perceived utility", but star ratings were read more than twice as fast as text. The authors concede, "While almost half the users prefer pure text recommendation or text combined with stars, the consumers who use the ratings have a high preference for text." Thus, there seems to be value in text over a simple rating score, since the text provides more informational depth.

Other research enables similar features to ours but utilizes non-extractive methods. Collaborative filtering, often used in recommender systems, identifies similar objects given one object's user ratings and structured metadata [2, 18]. Document similarity techniques such as the vector space model [17] treat text as a bag-of-words to retrieve similar documents. In contrast, REVMINER uses the extracted attributes and their distributions of values from the unstructured review text to identify similar restaurants. Our work also relates to semantic search [6], where the search system seeks a "shallow" understanding of the meaning in the query and documents, and use this understanding to provide better results. REVMINER also builds on shallow understanding of text in the extraction of attributes and values, and in its parsing of queries into attribute searches.

Faceted search is another method for navigating documents. It entails using metadata (what we extract from unsupervised text and call 'attributes') to browse a collection of documents [7]. Hearst et al. explore faceted search for browsing images, using attributes such as "material", "location", and "view". While we opt for a search interface rather than a browsing interface for the attribute-value extractions, the extraction technology in REVMINER can enable faceted search as well. Karlson et al. develop a faceted search interface on mobile devices, emphasizing the importance of a compact interface [10]. These techniques can potentially be used to navigate user reviews as well, complementing our work.

Finally, Hoffmann et al. [8] take a similar approach to ours but rather than extracting attributes and values from reviews, they extract programming keywords from source code and documentation. This allows developers to perform programming-related queries over computer programs.

EXTRACTION AND PROCESSING

REVMINER consists of an offline system that extracts information from a corpus of reviews (see Figure 2), and a user interface that enables users to query for restaurants based on the extracted information. The interface is described in the Mobile App section.

The offline system consists of the following key components:

1. **Attribute-Value Extractor:** The main module that generates extractions from a corpus of reviews.
2. **Polarity Computer:** Generates a table from Yelp review scores that associates each value with a score on the positive-negative scale.
3. **Attribute Clustering:** Clustering algorithm that groups together related attributes to form categories used in the Color Bar interface.
4. **Similarity Scoring:** Computes a similarity score between each pair of restaurants.

The offline system in REVMINER automatically learns, given a corpus of reviews, the set of attributes that are discussed (*e.g.*, dim sum, happy hour, service, *etc.*), their associated values (*e.g.*, fresh, outstanding, slow), the polarity of values (*e.g.*, "exquisite" is stronger than "good"), clusters of similar attributes (*e.g.*, food, which would contain pizza, tofu, and seafood), and similarity scores between restaurants. We now describe each of the components from the architecture diagram shown in Figure 2.

Learning to Extract Attributes and Values

Our goal was to design an approach for automatically inducing a set of attribute-value pairs, which we will call extractions, with no pre-existing bias toward the vocabulary that would be used.

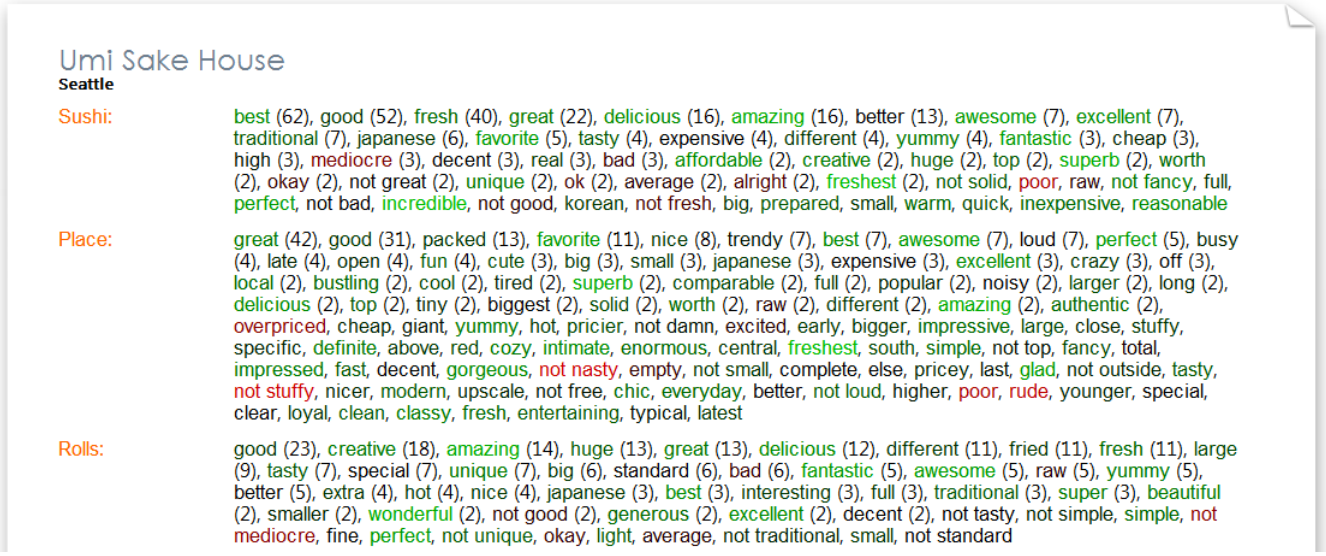

Figure 3: An example of the "raw" extracted values for the three most-frequent attributes for the Umi Sake House restaurant. The number beside the value represents how often that value has been used to describe the attribute in the reviews. A small subset of these values is presented to the user in the REVMINER app.

We developed an automated bootstrapping method, through empirical experimentation on a development corpus. This approach is based on earlier work on bootstrapping patterns for information extraction tasks [1, 3, 14, 15]. As we describe below, the redundancy in reviews provides powerful leverage for utilizing relatively simple extraction templates, and filtering the results to ensure quality.

The bootstrapping method is initialized with a small list of seed extractions that provide the initial cues for learning. These included, *food* as potentially *tasty*, *delicious*, or *disgusting*, and *service* as potentially *fast*, *helpful*, or *polite*. Given a set of extractions, the bootstrapping process repeatedly builds 'phrase templates'[1] that can, with high reliability, be used to expand the set of extractions. These templates include all of the words that occur between an attribute-value pair. For example, the phrase "parking here is terrible" would provide the template "[attribute] here is [value]" for the attribute *parking* with value *terrible*. We extract templates that commonly occur with arguments in our current extraction set and apply them to the complete corpus to grow the extraction set. Figure 4 shows pseudo-code for the full process. At each round, we (1) fix the extractions and update the templates, (2) fix the templates and values to extract new attributes, and (3) fix the templates and attributes to extract new values. This process is performed repeatedly, each time with higher frequency thresholds for whether the new templates or extractions can be included, until the set of extractions converges.

In general, each extraction comprises one word for an attribute with one word for the value. However, we sometimes allow multi-word extractions and multiple extractions per sentence or phrase. Two-word attributes are constructed when two nouns (identified by a part-of-speech tagger) appear together. This allowed REVMINER to include attributes such as *wait staff*, *wine list*, and *pad thai*. We also apply simple rules to identify boolean patterns in sentences. For example, REVMINER understands "and" as an operator over sentences such as "The service we received was both fast and friendly." This allows REVMINER to do multiple extractions, to extract *fast* and *friendly* as values associated with *service*. Finally, we extract multi-word values for negations that appear in templates by prepending the negative word to the extracted value. For example, for the sentence "Fries are not quite crispy" and the template "[attribute] are not quite [value]" we would extract the attribute *fries* with value *not crispy*.

We found during development that this bootstrapping provided many high-quality extractions but also over-generated by producing a much larger number of spurious extractions such as "guys = happy" for the "The guys were happy with the food." or "hand = tiny" for "My hand was tiny compared to the burger." This led us to refine the process with a number of filters. First, we expect extractions, across a large corpus, to appear with and without good patterns. For example, both "parking here is terrible" and "terrible parking" should be seen. If the more compact version is not seen for at least 5% of the extractions for a pattern, we discard the pattern. Additionally, we drop a number of qualifier words from the patterns, for example "lots" or "plenty," to avoid confusions they can cause. Finally, we used a part of speech tagger to ensure that in extractions with no pattern, such as "terrible parking," the first word is an adjective and the second is a noun. Together, these filters reduced the number of undesirable extractions shown in the extractive interface.

In total, 200 unique attributes and 1,536 unique values were extracted from the review data.

[1]These are sometimes called 'lexical patterns' in the literature.

Inputs: A large corpus of reviews R.

Definitions:

- Let E be a set of extractions. Each $e \in E$ is an attribute, value pair, for example $e = (a = \textit{parking}, v = \textit{terrible})$. Define $A(E)$ to be the set of attributes that exist in some pair in E and $V(E)$ to be the set of values in E. Let T be the set of extraction templates. One example $t \in T$ might be "[attribute] here is [value]." Define γ to be a count cutoff threshold.

Initialization:

- Create the empty template set $T = \emptyset$. Set ϵ based on dataset size ($\epsilon = 6$ for Yelp Seattle data). Set the cutoff to $\gamma = \epsilon$. Initialize E with a small set of seed extractions. For example, *food* paired with *tasty*, *delicious*, and *disgusting*.

Algorithm:

Do until the size of E does not grow:

Step 1: (Template Induction)

a. Find the set of templates T' that appear with attribute-value pairs in E somewhere in the corpus R. See text for full details.

b. Filter T' to remove templates that do not match the frequency threshold δ, and also as described in the main text.

c. Set $T = T'$.

Step 2: (Attribute Induction)

a. Let E' be every extraction from the corpus R for a template in T that contains a value in $V(E)$.

b. Filter E' according to the frequency cutoff δ.

c. Set $E = E'$.

Step 3: (Value Induction)

a. Let E' be every extraction from the corpus D for a template in T that contains an attribute in $A(E)$.

b. Filter E' according to the frequency cutoff δ.

c. Set $E = E'$.

Step 4: Update cutoff $\delta = \delta + \epsilon$.

Output: The final extraction set E.

Figure 4: REVMINER's bootstrapping algorithm for extracting attribute-value pairs from a large corpus of reviews.

Polarity

To determine the polarity of a value (*e.g.*, whether *delicious* is a positive or negative value), we use a simple method that works well on a large dataset such as ours. For each value, we compute the average of the Yelp ratings for all reviews it appears in. If this value is above or below empirically set thresholds, we label it as positive or negative. Otherwise, it is neutral. Across the potentially tens of thousands of reviews containing the value, this technique yielded high-quality polarity scores for nearly all values.

To evaluate the polarity generated from extracted values and their corresponding review ratings, we compared REVMINER and SentiWordNet (used in Review Spotlight [21]) to a gold standard of labels. SentiWordNet is a manually constructed resource that assigns positive and negative scores to subjective adjectives [5]. We selected the most appropriate word form from SentiWordNet based on the context and labeled it positive if the positive score was greater than the negative score, and negative if vice versa. This seemed to be a heuristic in which SentiWordNet performed well. Two judges labeled 199 values as either positive, negative, or neutral (with a weighted Kohen's kappa of $\kappa = 0.80$, representing high inter-rater agreement) and resolved differences. We then compared both REVMINER and SentiWordNet's labels with the judges' labels. REVMINER's labels agreed with the gold standard more often than SentiWordNet ($\kappa = 0.52$ for REVMINER, $\kappa = 0.48$ for SentiWordNet), demonstrating the effectiveness of our polarity computing method over this dataset.

Attribute Clustering

The Color Bar interface in Figure 6 displays attributes in five categories (*food*, *service*, *decor*, *overall*, other), each of which is a cluster that averages over a large number of primitive attributes. We manually selected the five clusters and seeded each with a small set of example attributes. For example, *decor* included *atmosphere* and *view*. We also define the distance between two attributes to be the Kullback-Leibler divergence [11] of the normalized count vectors for all values they appear with in the corpus. The clustering algorithm then simply places the remaining attributes in the cluster which has the closest seed attribute, according to this metric. This approach worked well for the small set of clusters we were interested in, outperforming other techniques such as k-means or agglomerative clustering in our development experiments.

Restaurant Similarity

We can use the extractions to compare different restaurants, rather than structured metadata or bag-of-words, as is often used. To determine which restaurants are most similar to a target restaurant, we iterate through each attribute, comparing their distributions of associated values. We experimented with a few metrics and found empirically that macro-averaging per attribute the Kullback-Leibler divergence of values returned the most intuitive results. The Kullback-Leibler divergence [11] (with a smoothing parameter) measures how far apart the restaurants' attributes are from each other. The attributes are weighted by how many values they are associated with, and summed together. The restaurants with the lowest divergence from the target restaurant are deemed the most similar and are presented in the REVMINER mobile app. For example, the two businesses most similar to Agua Verde Cafe (a traditional Mexican restaurant) were Cactus Restaurant and Rositas Mexican Grill.

Similarity based on fine-grained attributes automatically extracted from reviews differs from similarity based on Yelp metadata. Take for example, the restaurant called Paseo in the Fremont neighborhood. Yelp places it in the cuisine categories of Caribbean, Sandwiches, and Cuban. Its price range is $ for "Cheap, Under $10". Very different from Salumi Artisan Cured Meats in Pioneer Square, which Yelp categorized as Italian, Delis, Meat Shops with a $$ price range (*i.e.*, "Moderate, $11-$30"). Yet REVMINER identifies Salumi as its most similar restaurant to Paseo. Can that be right?

Careful inspection of the extracted attributes (Figure 5) in REVMINER for Paseo and Salumi finds the following: 78 mentions of a "long line" at Paseo, 76 mentions of a "long

Figure 5: An example where two restaurants *seem* unrelated based on Yelp's cuisine and price meta-data, but are noticeably similar when inspecting the finer-grained attributes extracted from reviews by REVMINER.

line" at Salumi; raving descriptions of the sandwiches and food at both restaurants, many mentions of the place being "small" as well as the seating being "limited" at Paseo and Salumi. From this, we can infer that these both attract lunch crowds where the long lines in front of small store with limited seating are worth the wait for delicious sandwiches. Thus, the metadata presented in the Yelp app's search results page suggests there is little similarity between the two restaurants, but a deeper inspection by the similarity scoring system identifies a surprising similarity based on the extractions. Of course, REVMINER also performs well in common cases of finding similarity by typical attributes relating to cuisine, foods offered, service quality, etc.

MOBILE APP

The REVMINER Android app enables users to search and learn about the restaurants mentioned in our review corpus. The app consists of three capabilities: *search* for restaurants by listing attributes of interest, *view* information about each restaurant via four compact interfaces (Figure 6), and *suggest* similar restaurants. Each capability is described below.

Attribute Search

The REVMINER app enables users to query by attribute, and view restaurant results ranked by the polarity, strength, and frequency of the extracted values for those attributes.

Given a query, REVMINER first identifies which terms are attributes and which terms are values, and then pairs each value with the closest preceding attribute. For example, the query "free parking cheap mexican food" is parsed as the attribute = value assignments: parking = free, food = cheap, and food = mexican. Now, given a set of assignments, restaurants are ranked according how often their review text mentioned that pair by matching the assignments with extractions.

For the above example query on the Yelp Seattle corpus, this approach would return Rancho Bravo, a local taco shop with a parking lot in the back, as the top result. These types of queries are difficult to process in a traditional information retrieval system which cannot identify attributes and values in the query, and relies on seeking the words in the document. Thus, a review noting "free food" would score well in traditional information retrieval systems even though that is not the intent in the query "free parking cheap mexican food".

REVMINER's attribute search also distinguishes between subjective and objective searches. When a user searches for "good service", they do not literally mean "service that people said was good". Instead, they mean "service that people said was good or better". Therefore, we treated some values such as *best*, *good*, *great*, *amazing* as subjective values, which switched the query into finding the restaurant that maximizes the associated attribute (based on the polarities of values for the restaurant's attribute). For example, if the query is "good margaritas", REVMINER will return restaurants where the margaritas are called "exquisite" and "amazing" ahead of ones where the margaritas are merely "good", even though the values are not a literal match to the query. For objective values such as *creamy* or *chinese*, this behavior would not be desirable.

Note that we could easily supplement attribute searches with traditional information retrieval techniques. For example, we could fall back to a bag-of-words technique like the vector space model if our attribute searches return no results. We have not done so to focus on assessing the behavior of a "pure" extractive interface.

Presenting Restaurant Information

Reducing large numbers of reviews into a set of extractions allows REVMINER to summarize a restaurant on a single interactive page. Rather than presenting dozens or hundreds of reviews to the user as on a typical review website, REVMINER lays out the aggregated extractions for a restaurant, allowing the user to see what attributes other users comment most frequently about for a restaurant, what opinions they held about particular attributes for the business, and which were the most common extractions.

For example, Umi Sake House is a sushi bar with over a thousand reviews listed across 40 pages. The extractions for Umi Sake House comprise 101 attributes with an average of 12 distinct values each. Many attribute-value pairs are mentioned numerous times like "best sushi" (62) and "great place" (42). This collapses the information substantially, trading 40 pages of text for a table of extractions. The extractions are then sorted and laid out differently depending on the interface currently chosen. Four distinct interfaces are offered in the REVMINER mobile app.

The default REVMINER interface (the **Common** interface) lists the attributes vertically, with the most frequent values associated with the attribute shown underneath. The attributes themselves are sorted in descending order based on the number of values associated with them. The values are color-coded according to their polarity and strength (the greener the more positive, the redder the more negative). This gives the user a sense of the most common things reviewers say about the restaurant's attributes. Users can scroll down to less frequent attributes using the familiar swipe-to-scroll gesture.

(a) Common (b) Special (c) Tag Cloud (d) Color Bars

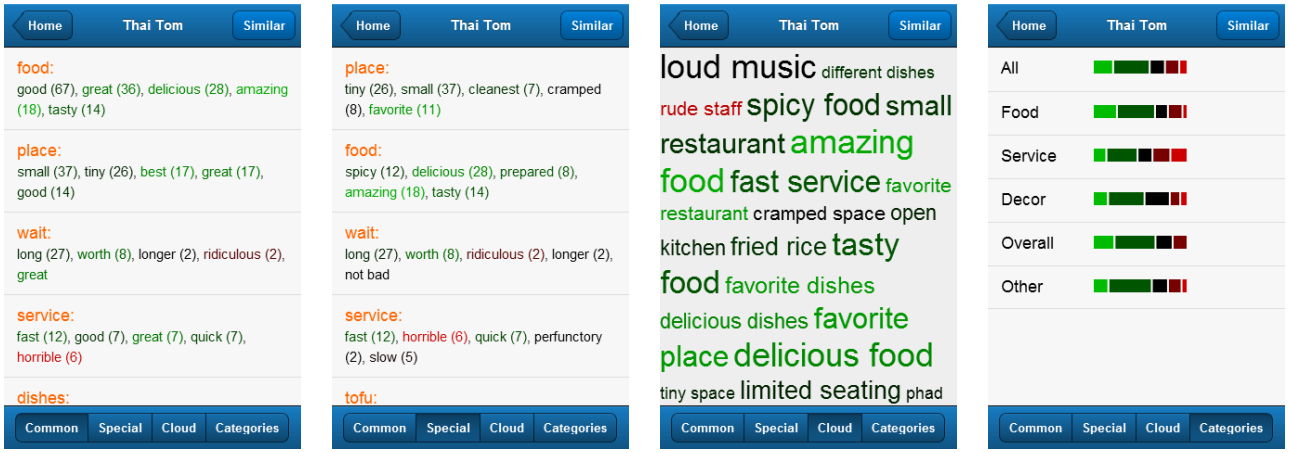

Figure 6: The four interfaces available to view information about Thai Tom in REVMINER. The default Common interface is a list of most-frequent attributes paired with their most-frequently-associated values. Compared to the Common interface, the Special interface selects more unusual values to show, the Tag Cloud interface lays out the attribute-value pairs differently, while the Color Bar interface clusters the attributes into categories and shows the values in a color-coded stacked bar chart.

Tapping on the attribute would show the source sentences containing that attribute, and tapping the sentence would show the entire review containing the sentence. This allowed users to zoom into a particularly interesting attribute.

The **Special** interface presents the attributes in the same manner, but sorts the values based on different criteria. The values are sorted not just by frequency, but also how unusual the value is. This resembles the tf-idf equation used in information retrieval systems [9], where the inverse of the term's frequency is multiplied by its frequency in the document (in our case, the reviews). We hypothesized that this presentation surfaced the more interesting aspects of a restaurant. For example, in Figure 6, key aspects of the restaurant become more prominent: the small tiny place, spicy food, and minority of customers who thought the service was horrible.

We implemented a **Tag Cloud** with some features similar to those from Review Spotlight[2] [21], which differed from a typical tag cloud by treating attribute-value pairs as a single term, placing them adjacent to each other in the cloud. The size of the words represented their frequency in the reviews. For example, "fresh sushi" in large font would indicate that many reviewers noted the sushi was fresh (possibly in multiple different lexical forms). Users could use this to gain an immediate impression of which attribute-value pairs were most pronounced in the reviews.

The final interface was **Color Bars** for attribute clusters, similar to those used in shopping websites like Google Shopping. Like Google Shopping, REVMINER's color-coded stacked bar charts presented the polarities drawn from the values for different attribute clusters (*e.g.*, food, service, decor). The color bars show at a glance the proportion of positive, neutral, and negative values describing each attribute. Clicking on the attribute cluster presented either a list (like the Common interface) or a Tag Cloud interface for that attribute cluster, depending on what was last selected by the user.

[2]Review Spotlight's tag cloud presented their pairs differently and included interactive features that REVMINER did not. Notably, the noun and adjective could be different sizes and were laid out more sparsely than a typical tag cloud. The user could also interact with the tag cloud—the source sentences were revealed by clicking the pairs, and adjectives for each noun were combined but were shown when hovering over the pair.

Similar Restaurants

Through the similarity scores computed in the offline processing, users had the option to view restaurants similar to the one they were already viewing. The app retrieved the five most similar restaurants and allowed the user to navigate to the information about those retrieved restaurants. From there, REVMINER would again retrieve the five most similar restaurants, thus allowing the user to browse similar restaurants indefinitely.

USER STUDY

We conducted a user study of REVMINER to address two key questions:

1. What is a compelling way to display the attribute-value information extracted by REVMINER on a smartphone?
2. What "natural" queries take advantage of an extractive interface over a search interface in the restaurant domain?

To answer the first question, we compared REVMINER's interface to three alternatives: a Tag Cloud interface which re-implements portions of the Review Spotlight interface, a Color Bar interface based on Google's shopping interface, and a Special interface, which highlights the distinctive values associated with each attribute (instead of the most common ones as in REVMINER's default interface). To answer the second question, we compared REVMINER's interface to Yelp's returned results on a set of attribute queries crawled from Twitter (described in the next subsection).

Method

Fourteen participants helped in evaluating REVMINER using two Android smartphones with the REVMINER app preinstalled. The participants were mostly recruited from mailing lists for graduate students; the recruitment email asked for participants who had lived in Seattle for at least two years, and frequently dined out. This allowed us to acquire participants who were familiar with many of the restaurants they would encounter in study. The study lasted approximately 90 minutes per participant, and we compensated them with snacks and a gift card drawing.

For the **interface** task, each participant was asked to list 10 restaurants they were familiar with. They looked up each restaurant in REVMINER and were presented with the four interfaces described earlier. The participant then studied and interacted with each interface; sometimes, the participants would use both phones provided to them for side-by-side comparisons. The participants rated the interfaces based on how well the interface accurately reflected what they knew about the restaurant, and whether it was useful for presenting attributes about the restaurant.

For the **search** task, participants evaluated how well REVMINER compared to Yelp in returning restaurant results related to a search query. REVMINER was loaded on one phone, while the Yelp app was loaded on another phone with geolocation features disabled. Therefore, the geographical scope of REVMINER and Yelp were the same (Seattle area), and the underlying dataset was the same, except Yelp's dataset had an additional 8 months of data accrued after the development of REVMINER.

We collected actual questions people had about restaurants automatically by searching Twitter. First, we copied 48 messages (originally 50, but 2 were discarded as irrelevant to restaurants) that contained "What restaurant has" or some variation of that phrase, from public Twitter messages. These messages typically asked for recommendations of restaurants with particular attributes, specifying an information need. For example, "What restaurant makes the best mexican food in Seattle?" Participants were then asked to formulate queries that could satisfy the information need from a selection of 10 such messages. They entered the query into the search box of REVMINER and Yelp, and rated the quality of the ranked list of results, providing us with a comparison between REVMINER and Yelp.

We examined the results from the search task and noticed that REVMINER performed particularly well for queries that contained more than one attribute. For example, one information need was "Which restaurant has the best wings for best price?" Queries containing multiple attributes took advantage of the attribute association and scoring mechanism used with the REVMINER extractions. However, since these queries only comprised a small portion of the original evaluation, we asked our last four participants to evaluate the efficacy of these 'conjunctive queries'. Four of the 48 Twitter messages were conjunctive information needs, and these last participants formulated and evaluated queries for each of the four conjunctive information needs.

Table 1: REVMINER's default interface compared with three alternatives in a user study. REVMINER's Common interface is preferred over Tag Cloud, Color Bar, and Special by a statistically significant margin.

Comparison	Yes	No	Tie	p-value
Common > Tag Cloud	46%	20%	33%	< 0.05
Common > Color Bar	73%	10%	18%	< 0.05
Common > Special	54%	14%	32%	< 0.05

Results

We present here the quantitative results for the two tasks and our qualitative interpretations of them.

Our first task resulted in 114 **interface** comparisons from the 14 participants (some participants did not perform the full 10 comparisons due to time constraints).

Comparing Common to Special: Both interfaces have the same layout, and thus can show the same amount of information on the screen. The difference is in the values chosen to display—whether REVMINER displayed the most frequent values for each attribute, or gave preference to the unusualness of the values as well. In our study, we found that participants rated Common higher in 54% cases, Special higher in 14% cases, and rated them equally in 32% cases. Thus, while there were several cases where unusual values were useful as they highlighted aspects of the restaurant that deviated from the norm, in most cases, they were outliers and detracted from the understanding of the restaurant.

Comparing Common to Tag Cloud: Both interfaces choose the content to show in order of frequency, but use a different layout. Common presents a list of attributes along with their values, while Tag Cloud is a similar implementation to the tag cloud in Review Spotlight [21]. Participants preferred Common to Tag Cloud in 46% cases, while the inverse was true in 20% cases, and participants rated them equally in 33% cases. One participant reasoned that the Common interface was able to pack more information in the same screen space, while being faster to glance. However, other participants did prefer the Tag Cloud because the most important terms "popped out of the screen" in large font. We found this comparison to be the most polarizing among participants.

Comparing Common to Color Bar. We experimented with adding an additional layer of abstraction above the Common interface. The stacked bar charts for each category, which comprised multiple related attributes, showed the distribution of polarities for the values in that category. Participants preferred Common 73% to 10% of the time, with 18% having same ratings. Participants rating Common higher explained that many restaurants had similar bar charts, and it was difficult to compare between bar charts. They preferred having descriptive values to understand the attributes in a richer way.

We used a two-tailed Sign Test to test for differences in the $N = 114$ interface comparisons. We found the differences between Common and the other three interfaces were statistically significant at the $p < 0.05$ level after applying the Bonferroni correction.

Table 2: REVMINER's restaurant results compared with restaurant results in the Yelp mobile app.

Result	Frequency
REVMINER Favored	45%
Yelp Favored	30%
REVMINER and Yelp Tied	25%

In the **search** comparison between Yelp and REVMINER, participants performed 115 comparisons (not all participants did 10 comparisons due to time constraints). The findings showed that in 9 cases (8% of queries), REVMINER produced no results. This was due to terms in the query not matching any of the attributes of values extracted by REVMINER (*e.g.*, mac n cheese). Yelp's bag-of-words representation enables it to produce some results for these queries. For the rest of the queries, participants preferred REVMINER in 45% cases and preferred Yelp in 30% cases. Yelp and REVMINER scored the same in the remaining cases.

Upon closer inspection of the queries where REVMINER produced substantially better results, we found that conjunctive needs were a particular strong point for REVMINER. We asked our last four participants to select the Twitter questions with conjunctive information needs to focus on evaluating these. For the conjunctive queries, participants preferred REVMINER more than twice as often as Yelp (62% preferring REVMINER, 27% preferring Yelp). This suggests that an extractive approach may perform better than the traditional information retrieval approach that Yelp uses for conjunctive queries. One explanation is that a system understanding how the values are associated with different attributes can identify relevant restaurants better than a bag-of-words approach to query-restaurant matching.

DISCUSSION

Contrasting with Information Retrieval

Extractive interfaces such as REVMINER differ from traditional information in a few ways. First, extractive interfaces can treat words as ordinal values. This means understanding that *great* and *good* are both subjective values, and that *great* is more positive than *good*. Thus, values can be placed on a scale and compared. Second, extracting the attribute-value pairs allow an extractive interface to aggregate attributes and values in many ways. This enables applications such as clustering and similarity which depend on the distributions of aggregated values. Furthermore, documents (restaurants in our case) can then be ranked based on a scoring function using the aggregated query terms. Third, values are associated with attributes when appropriate, and thus the query and document are not treated as a bag of words, as is typical in information retrieval. The query "Free parking amazing desserts" will not match reviews containing mentions of "free desserts" and "amazing parking". Finally, we able to take advantage of boolean-like operators in sentences such as "not" and "and" to generate more extractions.

Meaning in Review Language

During the processing of extracting attributes and values from the reviews, we learned two lessons about meaning inherent in the language used in reviews. First, words can have different meanings when it is written in a review, versus when it is issued in a query. For example, people who mention that the food at a restaurant is *cheap* does not necessarily mean it cost little. What they mean is the food is good value compared to their expectations. Few people will remark that McDonald's was *cheap*, but people will declare *cheap* over a $30 3-course special at an exclusive establishment. But when searching for "cheap restaurant", the same people seek cheap eats in absolute terms, rather than somewhat cheaper than they would typically expect. An edge case such as this can cause problems for both traditional information retrieval and extraction-based search.

The meaning in a query itself can also be ambiguous. Someone looking for "good dim sum" is not looking for dim sum where people have noted is "good". In fact, our polarity calculations show that "good" is barely a positive complement in reviews. Instead, the searcher is looking for "good or better" dim sum, including dim sum that others have described as "great" or "amazing". Not all values are subjective in this way, as others such as "spicy" or "italian" describe a specific quality in the attribute. REVMINER is able to understand this and by identifying subjective values such as *good* in the query, to give the searcher what they really want: dim sum restaurants sorted by quality.

Extending Similarity

Item similarity typically uses a collaborative filtering algorithm such as k-nearest neighbors applied to user browsing data or user ratings. As part of REVMINER, we have developed a good similarity scoring function based solely on user reviews. Thus, as an independent source of data, user reviews can be combined with other user data to develop even more accurate similarity-based applications.

Beyond Restaurant Reviews

There is a growing dataset of reviews on the Web, such as product reviews and movie reviews. While we have used REVMINER on other types of non-restaurant reviews business reviews, non-business reviews may pose other interesting challenges. For example, movie reviews often contain more descriptions of the movie content, but some attribute-value extractions are still quite useful, such as "surprise ending", "predictable plot", "poor acting". Mining the reviews to produce similarity scores may be useful as a supplement to existing methods.

User-Review Behavior

Our research presents a new opportunity for users of review websites. It is currently common to read reviews after finding a particular business or product, rather than using reviews to discover new products. This may be due to the large amount of review text that one has to wade through. An extractive interface may enable a change in this behavior, by presenting aggregate review information as extractions and enabling better searching capabilities in reviews. Perhaps this may change peoples' behavior so that reading reviews might not remain the last action in a shopping process. Instead, people may go to the reviews first to find a particular business or product matching their needs.

Limitations

We acknowledge some limitations to our evaluation of REVMINER. In evaluating the extractive interface, we focused on the choice of which extractions to present and how to present them. We did not compare the interface with the full Yelp mobile app because the Yelp app also displays metadata such as photos, the address, opening hours, etc. which were beyond the scope of this paper. Additionally, we did not perform rigorous counterbalancing of the order we presented the interfaces to the participants. Participants browsed the four extractive interfaces on two phones informally, sometimes side-by-side, sometimes one after another.

CONCLUSION

This paper described a method for extracting attribute-value pairs from a large corpus of user reviews. These extractions provide a structured representation of the opinions people have about restaurants and their attributes. Using the extractions, we developed REVMINER, which enables searching based on attributes, presenting information for restaurants, and identifying similar restaurants. We created an Android app, which offers four compact presentations of the extracted information, and carried out a user study.

Our user study helped to quantify the advantage of the extractive interface over both earlier research and over Yelp's smartphone app. We believe that our experiments, and the REVMINER system demonstrate the potential of extractive interfaces to enable a better experience for shoppers navigating product reviews on smartphones.

ACKNOWLEDGMENTS

This research was supported in part by NSF grant IIS-0803481, ONR grant N00014-08-1-0431, a gift from Google, and carried out at the University of Washington's Turing Center. The authors thank James Fogarty for helpful discussions about study design, and Dan Weld and Raphael Hoffman for reviewing earlier drafts of the paper.

REFERENCES

1. Agichtein, E., and Gravano, L. Snowball: Extracting relations from large plain-text collections. In *Proceedings of DL* (2000), 85–94.
2. Breese, J., Heckerman, D., and Kadie, C. Empirical analysis of predictive algorithms for collaborative filtering. In *Proceedings of UAI* (1998), 43–52.
3. Brin, S. Extracting patterns and relations from the world wide web. *The World Wide Web and Databases* (1999), 172–183.
4. Chang, C.-H., Kayed, M., Girgis, M. R., and Shaalan, K. F. A survey of web information extraction systems. *IEEE Trans. on Knowl. and Data Eng. 18*, 10 (Oct. 2006), 1411–1428.
5. Esuli, A., and Sebastiani, F. Sentiwordnet: A publicly available lexical resource for opinion mining. In *Proceedings of LREC*, vol. 6 (2006), 417–422.
6. Guha, R., McCool, R., and Miller, E. Semantic search. In *Proceedings of WWW* (2003), 700–709.
7. Hearst, M., Elliott, A., English, J., Sinha, R., Swearingen, K., and Yee, K.-P. Finding the flow in web site search. *Commun. ACM 45*, 9 (Sept. 2002), 42–49.
8. Hoffmann, R., Fogarty, J., and Weld, D. S. Assieme: finding and leveraging implicit references in a web search interface for programmers. In *Proceedings of UIST* (2007), 13–22.
9. Jones, K. S. A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation 28*, 1 (1972), 11–20.
10. Karlson, A. K., Robertson, G. G., Robbins, D. C., Czerwinski, M. P., and Smith, G. R. Fathumb: a facet-based interface for mobile search. In *Proceedings of CHI* (2006), 711–720.
11. Kullback, S., and Leibler, R. A. On information and sufficiency. *Ann. Math. Statist. 22*, 1 (1951), 79–86.
12. Pang, B., and Lee, L. Opinion mining and sentiment analysis. *Found. Trends Inf. Retr. 2*, 1-2 (Jan. 2008), 1–135.
13. Popescu, A.-M., and Etzioni, O. Extracting product features and opinions from reviews. In *Proceedings of HLT* (2005), 339–346.
14. Probst, K., Ghani, R., Krema, M., Fano, A., and Liu, Y. Semi-supervised learning of attribute-value pairs from product descriptions. In *Proceedings of IJCAI* (2007), 2838–2843.
15. Riloff, E., and Jones, R. Learning dictionaries for information extraction by multi-level bootstrapping. In *Proceedings of AAAI* (1999), 474–479.
16. Rivadeneira, A. W., Gruen, D. M., Muller, M. J., and Millen, D. R. Getting our head in the clouds: toward evaluation studies of tagclouds. In *Proceedings of CHI* (2007), 995–998.
17. Salton, G., Wong, A., and Yang, C. S. A vector space model for automatic indexing. *Commun. ACM 18*, 11 (Nov. 1975), 613–620.
18. Su, X., and Khoshgoftaar, T. M. A survey of collaborative filtering techniques. *Adv. in Artif. Intell. 2009* (Jan. 2009), 4:2–4:2.
19. von Reischach, F., Dubach, E., Michahelles, F., and Schmidt, A. An evaluation of product review modalities for mobile phones. In *Proceedings of MobileHCI* (2010), 199–208.
20. von Reischach, F., Guinard, D., Michahelles, F., and Fleisch, E. A mobile product recommendation system interacting with tagged products. In *Proceedings of PerCom* (2009), 1–6.
21. Yatani, K., Novati, M., Trusty, A., and Truong, K. N. Review spotlight: a user interface for summarizing user-generated reviews using adjective-noun word pairs. In *Proceedings of CHI* (2011), 1541–1550.