OpinionBlocks: A Crowd-Powered, Self-improving Interactive Visual Analytic System for Understanding Opinion Text

Mengdie Hu[1], Huahai Yang[2], Michelle X. Zhou[2], Liang Gou[2], Yunyao Li[2], and Eben Haber[2]

[1] Georgia Institute of Technology
mengdie.hu@gatech.edu
[2] IBM Almaden Research Center
{hyang,mzhou,lgou,yunyaoli,lgou,ehaber}@us.ibm.com

Abstract. Millions of people rely on online opinions to make their decisions. To better help people glean insights from massive amounts of opinions, we present the design, implementation, and evaluation of OpinionBlocks, a novel interactive visual text analytic system. Our system offers two unique features. First, it automatically creates a fine-grained, aspect-based visual summary of opinions, which provides users with insights at multiple levels. Second, it solicits and supports user interactions to rectify text-analytic errors, which helps improve the overall system quality. Through two crowd-sourced studies on Amazon Mechanical Turk involving 101 users, OpinionBlocks demonstrates its effectiveness in helping users perform real-world opinion analysis tasks. Moreover, our studies show that the crowd is willing to correct analytic errors, and the corrections help improve user task completion time significantly.

Keywords: Text analytics, text visualization, self-improving, crowd-sourcing.

1 Introduction

Hundreds of millions of people voice their opinions online daily. Large portions of these opinions are product reviews about "experienced goods"—products or services of which characteristics are difficult to observe in advance but can be learned after purchase [21]. Not only do product reviews provide great value to individual consumers and influence their purchasing decisions [24], but they also impact the product or service strategies of businesses [32]. However, gaining insights becomes increasingly challenging for users as the number of reviews gets larger and larger [7, 12, 13, 31].

To help users wade through a large number of reviews, commercial sites often employ one of two approaches. One approach, used by sites such as Amazon.com, lets readers vote on the helpfulness of each review, and directs future readers to the most helpful reviews. The other approach, applied by sites such as Bing Shopping and Google Product Search, provides an overview of the most frequently mentioned product/service features, and the overall sentiment expressed in a collection of reviews. Users can then filter the reviews based on the identified features.

P. Kotzé et al. (Eds.): INTERACT 2013, Part II, LNCS 8118, pp. 116–134, 2013.

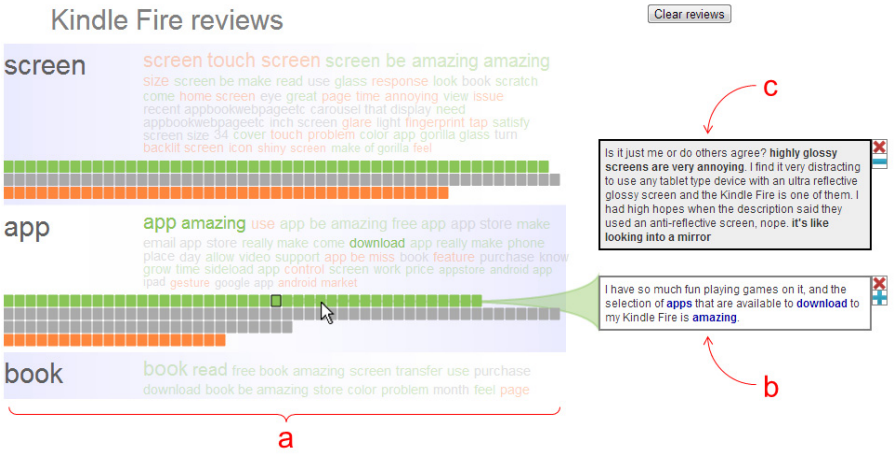

Fig. 1. The interface of OpinionBlocks: (a) a system-generated aspect-based summary; (b) a system-extracted review snippet; (c) the full text of a review.

While a few high-quality reviews or the aggregated sentiment may provide useful information, previous research shows that users often desire finer-grained understanding of the reviews [7, 31]. In particular, people process information in an attribute-driven manner in the absence of actual products (e.g., online shopping) [15]. In such cases, people examine the attributes of a product to evaluate whether the product fits their purchase goal (e.g., buying a camera for underwater adventures). In addition, the positive or negative sentiment expressed by the reviewers toward *each* attribute helps justify the suitability of the product [31].

To facilitate attribute-driven evaluation of products, a number of systems produce an aspect-based summary, including the extraction of sentiment toward each of the aspects [7, 13, 16, 31]. Among these systems, several recent ones use noun-adjective pairs to summarize the aspects of a product/service (noun) and the sentiment (adjective) toward each aspect [13, 31]. However, this approach has several limitations. First, they cannot handle implicit opinions. For example, they cannot extract aspects "*weight*" or "*size*" implied by the expression "*it is light and portable*" [31]. Second, they do not deal with conflicting opinions expressed by different reviewers. For example, one reviewer raves "*the screen is fantastic*", while the other complains "*positively claustrophobic in terms of screen usage*". In such cases, it is unclear whether multiple noun-adjective pairs would be displayed or one noun (e.g., "*screen*") would be associated with multiple adjectives (e.g., "*fantastic*" and "*claustrophobic*"). Third, the performance of these systems is limited by the imperfections in the underlying natural language processing (NLP) techniques. Because of the flaws in NLP (e.g., classifying "impeccable" as a negative sentiment), users may find certain summaries mystifying [31].

To improve the quality of aspect-based opinion summarization, researchers have developed sophisticated NLP techniques for aspect extraction and sentiment analysis

(see Section 2.2). However, due to the challenging nature of the problems, even with a large amount of domain-specific training data, state-of-the-art NLP techniques can only achieve 50% to 85% accuracy for either of these tasks. The imperfections in text analytic results often lead to user frustrations and even distrust in the system [31].

To address the challenges mentioned above, we have developed a novel interactive visual analytic system, OpinionBlocks, to meet two design goals: (1) automated creation of an aspect-based, effective visual summary to support users' real-world opinion analysis tasks, and (2) support of user corrections of system text analytic errors to improve the system quality over time. On the one hand, meeting the first goal motivates users to correct system errors. On the other hand, achieving the second goal improves the system quality, which then better aids users in their tasks.

To achieve the first goal, OpinionBlocks employs advanced NLP technologies to automatically create and present users with a fine-grained, aspect-based visual summary of opinions. As shown in Figure 1, the created visual summary allows a user to gain insights into a collection of reviews at multiple levels:

1. Frequently mentioned aspects of a product/service, including those *explicitly* and *implicitly* expressed in the reviews (Figure 1a).
2. The description of each aspect in a form of key phrases, a set of associated review snippets, and the inferred sentiment of each key phrase and snippet (Figure 1b).
3. The full review containing extracted aspects (Figure 1c).

To achieve our second design goal, OpinionBlocks allows users to interact with the visual summary to amend analytic errors (Figure 2). It then aggregates user contributions to update and improve the visual summary for future users.

To demonstrate the effectiveness of OpinionBlocks in meeting both design goals, we conducted two crowdsourced studies on Amazon Mechanical Turk involving 101 users for the analysis of 18,000 reviews of the Amazon Kindle Fire (2.8 million words). Our results show that more than 70% users successfully accomplished non-trivial opinion analysis tasks using OpinionBlocks. These tasks involve answering questions beyond the capability of existing systems, such as "*What is the most common use of the product*?" and "*Which aspect received most conflicting reviews?*". Furthermore, our studies show that users are not only willing to use our system to correct text classification mistakes, their corrections also produce high quality results. The participants in our study successfully identified many mistakes and their aggregated corrections achieved 89% accuracy. Incorporating the crowd corrections, OpinionBlocks is also able to help users significantly improve their task completion time. As a result, OpinionBlocks offers two unique contributions:

- It supports real-world, opinion analysis tasks beyond that of existing visual opinion analysis systems.
- It leverages the power of the crowd to self-improve the quality of the text analytic results and compensate for the limitations in today's NLP technologies.

In the rest of the paper, we present the details of OpinionBlocks after an overview of related work. We then describe our two crowdsourced studies and their results. Finally we discuss limitations and implications of our work before concluding.

2 Related Work

Our work is related to four main areas of work across HCI and text analytics.

2.1 User Interfaces for Understanding Opinion Text

To better help users extract insights from a large number of online reviews, researchers have developed various interactive systems. For example, Faridani et al. created Opinion Space, an interactive tool that allows users to visualize and navigate collected opinions [9]. Yatani et al. developed Review Spotlight, which presents a word-cloud summary of online reviews in noun-adjective pairs [31]. Carenini and Rizoli built a multimedia interface that facilitates the comparison of different reviews [7]. More recently, Huang et al. presented RevMiner, an interactive system that summarizes reviews in noun-adjective pairs to be presented in a compact mobile phone interface [13]; and Rohrdantz et al. designed a visualization system that supports feature-based sentiment analysis of time-stamped review documents [26]. Similar to these efforts, our work aims at creating summaries of online opinions to help users in their decision-making processes. However, we go beyond existing systems to help users answer more complex analytical questions, such as identifying the aspects with the most conflicting reviews. While these systems are limited by the NLP techniques they employ, OpinionBlocks also leverages crowd input to compensate for its deficiencies in text analysis and improve its quality over time.

2.2 Opinion Mining and Sentiment Analysis

To help users digest massive amounts of online reviews, an active research area intersecting NLP and machine learning is known as opinion mining. State-of-the-art opinion mining technologies automatically extract aspects discussed in the reviews, and identify the sentiment expressed toward each aspect. Comprehensive reviews of the field can be found in [23, 17, 18], and these works point out limitations with aspect-based sentiment analysis: existing techniques are yet to go beyond parsing relatively simple sentence structures and modeling sentiment words (often domain specific) within sentences. And few can handle implied opinions well. Because of these difficulties, even the latest work published in this field can only achieve accuracy scores ranging from 50% to 85% for only aspect extraction or sentiment analysis alone, depending on the domain and training data [14, 17, 18, 19, 20, 29]. Moreover, achieving this level of accuracy often requires large amounts of training data (e.g., labeled sentences indicating aspect and sentiment expressions), which is often difficult and costly to obtain, especially when covering reviews for a diverse set of products [17]. While OpinionBlocks employs state-of-the-art opinion mining techniques to create an initial summary of reviews, it supports user interactions to rectify the imperfections in machine-generated summaries and improve the summary results over time. Furthermore, our implicitly crowd-sourced user inputs become valuable training data for the NLP community [28].

2.3 Interactive Machine Learning

Our work is also related to research efforts in interactive machine learning, where a machine learning process is augmented by human intelligence to improve the results. For example, Patel et al. presented a development environment that helps developers find and fix bugs in machine learning systems [25]. Amershi et al. developed systems that can iteratively learn the desired results based on end-user interaction behavior [3, 4]. Similar to these efforts, our work also aims to improve machine intelligence (i.e., text analysis results) through user interaction. However, while prior work focuses on leveraging individual users to improve machine learning, we focus on leveraging the wisdom of the crowd to improve analysis of unstructured text, which presents unique challenges as described later (e.g., reconciling crowd inputs).

2.4 Crowd-Powered Systems

Since OpinionBlocks is designed to leverage the crowd to identify and amend system imperfections in text analytics, it is related to an emerging research area on creating crowd-powered systems. This new class of software systems combines machine and human intelligence to solve problems that are extremely difficult or impossible for either approach alone. For example, Soylent guides the crowd on Amazon Mechanical Turk to rewrite and shorten text on demand [5]. n.fluent employs both machine translation and online crowd to help translate documents[1]. Carlier et al. combines content analysis and crowdsourcing to optimize the selection of video viewports [8]. While existing crowd-powered systems explicitly solicit a crowd's help (e.g., via Amazon Mechanical Turk) and use the results to help others, OpinionBlocks leverages its own users as the crowd implicitly, and motivates them to perform tasks that ultimately benefit both themselves and others (e.g., correctly identifying both positives and negatives of a product aspect).

3 OpinionBlocks

OpinionBlocks is a web-based system with three key components: a visual interface, a text analytic component, and a user feedback integration component. Below we describe each of the components, including our design rationales.

3.1 Interactive Visualization

The visual interface is designed to support two main user tasks: interacting with the generated visual summary and the original reviews, and correcting system errors in text analytics.

[1] https://www.ibm.com/developerworks/mydeveloperworks/blogs/c7f41400-4eb9-477c-b6fb-042466407259/?lang=en

Visual Features to Support User Decision Making

OpinionBlocks aims at aiding users in their information-driven decision-making processes. Based on previous research [31, 13, 15] and our own informal user studies (interviews with 10 colleagues who recently made a major purchase), we learned that a user's first step is to gain an overall impression of the important aspects of a product from available information. Our visual interface thus consists of two main parts. As shown in Figure 1, the left panel displays a visual summary of all the major aspects extracted from a set of reviews. The right panel is initially empty but shows relevant review snippets as a user interacts with the visual summary on the left.

A generated visual summary is made up of a set of aspect blocks (Figure 1a). From top to bottom, the aspects are ordered by their number of mentions in a review collection[2]. Each aspect block further consists of three parts: (1) the aspect name, (2) a text cloud of keywords and phrases describing the aspect, and (3) a set of colored squares, each of which represents a review snippet describing the aspect. Automatically extracted from a review document (see below), a *review snippet* includes a sentence that expresses opinions toward the aspect. Three colors are used to encode the sentiment expressed in a snippet: green (positive), gray (neutral), and orange (negative). The words and phrases in the text cloud are extracted from the snippets, and are colored based on the aggregated sentiment orientation of the relevant snippets. The colored squares are placed in different rows by their sentiment orientation, facilitating the comparisons of contrasting sentiments in each aspect (e.g., how many positive versus negative comments for the *Screen* aspect?) and across all the aspects (e.g., which aspect received most conflicting reviews?).

Our design is motivated by previous research and our own study that review readers tend to form and adjust their impression of opinions by looking for most discussed and most debated aspects, and they tend to verbalize their impression with short descriptive phrases [31]. Thus we designed the colored snippet boxes to support explicit comparison of comment frequency and polarity of sentiment under different aspects. And we help users highlight review snippets by keywords and phrases (Figure 2 Left).

Furthermore, readers often wish to see the concrete evidence behind the extracted aspects and sentiment in a summary [16, 26]. OpinionBlocks enables users to "drill-down" through clicking or hovering on the visual elements, allowing them to see snippets associated with blocks, keywords associated with snippets, snippets associated with keywords, or even the full context of the original reviews (Figure 2 Right).

[2] Unlike existing systems, which count how many reviews contain an aspect, we compute how many *sentences* refer to an aspect. We thus decided not to show the count to avoid potential confusion.

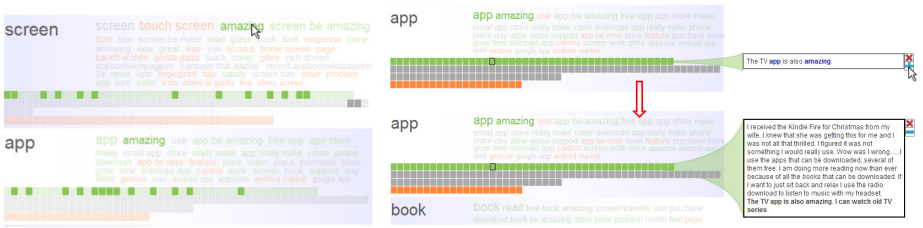

Fig. 2. Left: Hovering over the keyword "amazing" highlights all snippets containing the word. Right: Clicking on the "+" button next to a review snippet brings up the full review.

Interactive Features to Support User Feedback

As discussed earlier, one of our main design goals is to leverage the power of the crowd to identify and correct system errors, in particular, NLP errors that occurred in review analysis and summarization. Yatani et al. [31] suggests that showing the contextual text behind the phrases and sentiment classification helps compensate for the imperfect analytic results. However, we wish to take a step further and encourage the users to identify and correct text analytic errors. By correcting the errors, users not only obtain a more accurate visual summary for themselves, but also help future users of the system. We have identified four major types of system errors:

1. *Snippet omissions:* snippets that contain an opinion but were not extracted.
2. *Erroneous snippet extraction:* snippets without meaningful opinions
3. *Erroneous aspect:* snippets classified with the wrong aspect
4. *Erroneous sentiment:* snippets associated with the wrong sentiment

OpinionBlocks focuses on leveraging users to fix the last three types of errors, since identifying the first type of errors would require the users to be familiar with the entire review corpus. To rectify the errors, users can drag the colored square representing a misclassified snippet to the correct aspect or sentiment row (Figure 3 Left), or to somewhere out of the display area entirely if the snippet contains no meaningful opinion. Users can also click on an aspect name and change it to something more appropriate (Figure 3 Right).

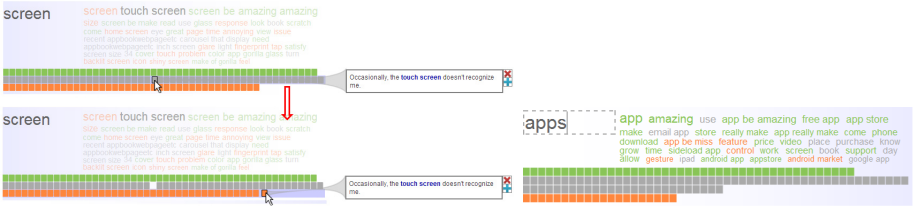

Fig. 3. Left: Moving a snippet misclassified as "neutral" to the "negative" row. Right: Changing the name of an aspect.

3.2 Opinion Mining and Sentiment Analysis

To generate the information used in our visualization, OpinionBlocks performs a four-step process: 1) review snippet extraction, 2) aspect extraction, 3) keyword extraction, and 4) sentiment analysis.

Review Snippet Extraction

From a collection of reviews of a product, OpinionBlocks extracts a set of review snippets that describe various aspects of the product. To extract a review snippet, OpinionBlocks first uses the OpenNLP parser [22] to obtain a parse tree for each sentence in a review. It then builds subject-verb-object (SVO) triples based on the parse tree. For each SVO triple, it checks whether the lemma of the verb matches a selective list of verbs (e.g., be, look, appear, etc.) from VerbNet [27], which are often associated with various aspects mentioned in a review. If there is a match, OpinionBlocks then keeps the subject of a SVO triple as an aspect candidate and the sentence containing the SVO triple as a review snippet. For example, given a sentence "*The display is made of Gorilla Glass, which is highly damage resistant*", the extracted SVO triple is: [*the display*, *make*, *Gorilla Glass*]. The sentence itself is a review snippet, and the subject "*the display*" then becomes an aspect candidate.

Note that we generate aspect candidates by considering only noun phrases that are also subjects of a restrictive subset of sentences in the review texts (by requiring their verbs to match a limited list). This approach is inherently resistant to noise introduced by common contextual information, such as prepositional phrases and discussions irrelevant to product aspects (e.g., detailed life experience like "*I tried out several different magazines*").

Aspect Extraction

Aspect extraction is to identify frequent *n*-grams from aspect candidates. Specifically, we first tokenize each aspect candidate and lemmatize its tokens with the Stanford Natural Language Processing Package [2]. Next, we extract all possible *n*-grams of size 3 from each candidate (or the candidate itself, if its length is shorter than 3), remove any stop word at the beginning or end of the n-grams, and calculate the frequency for each unique *n*-gram. Our preference of longer *n*-grams (e.g. tri-gram vs. bi-gram) is intentional: we observed that longer *n*-grams are typically more informative than shorter ones and thus are better at conveying concrete information to users. We then select and use the top-*K* (*K* is adjustable in our system) most frequent *n*-grams as a set of extracted aspects to summarize a collection of reviews.

We conducted several experiments to investigate whether our approach of aspect extraction can generate a consistent set of aspects given different sizes of the review collections. Here, we used the top-*K* aspects with the full review collection as the base line to investigate the performance of our approach with different sample ratios. Two metrics are employed here: Spearman's rank correlation coefficient (rho) [10] and coverage rate, where *rho* measures the correlation of two ranks of top-*K* aspects, and *coverage rate* measures the fraction of the top-*K* aspects from the full collection that

also occur in the top-*K* aspects from the subset of the collection. The two metrics were computed using twenty sample ratios. We performed ten test runs for each sample ratio and averaged the two metrics over the ten runs.

Figure 4 shows our experiment results. On the left, all reported *rho* values are over 0.8, which indicates that the top-*K* aspects identified with the samples are positively correlated to the aspects identified with all reviews (all values are significant). We also find that even with a small sample ratio of 0.35, the top-10 aspects have the exactly same rank as those identified using the full collection. The performance for top-20 and top-30 aspects with our approach is also very promising. For coverage rate, with a sample ratio of 0.35, our approach yields very good coverage (> 0.95) for top-10 aspects and around 0.8 for top-20 and top-30 aspects. As a result, our aspect extraction generates consistent results over different sizes of review collections.

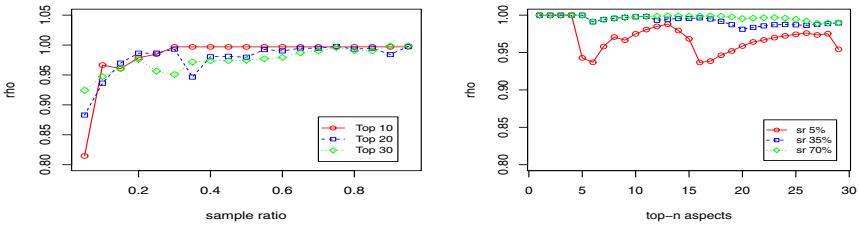

Fig. 4. Left: Spearman's rank correlation coefficient (rho) of top-K aspects with different sample ratios. Right: Coverage rate of top-K aspects with different sample ratios.

Keyword Extraction

To enrich an aspect-based summary, we also extract keywords from the relevant snippets for each aspect. We identify *n*-grams (unigrams, bigrams and trigrams) of all words and their frequencies from a collection of snippets related to an aspect. The *n*-grams with high frequency are used as keywords to describe each aspect. Because such a keyword relates to both an aspect and a snippet, we also use these keywords to index snippets. This way, keywords can be highlighted and easily spotted in a snippet when a user examines the snippet associated with an aspect (Figure 1b).

Sentiment Analysis

We use a simple lexicon-based approach [12] to infer the sentiment expressed in each snippet. This approach uses a public sentiment lexicon of around 6800 English words [1] to determine word sentiment orientation (positive or negative). We first tokenize and lemmatize a review snippet, and remove all stop words. The polarity of a snippet *s* is decided by its sentiment score $S(s)$, where $S(s)$ = |positive words in *s*| - |negative words in *s*|. If $S(s) > 0$, then *s* is considered positive; if $S(s) = 0$, then *s* is neutral; otherwise *s* is negative. If the verb of a SVO-triple contained in a snippet is associated with negation (e.g. "is not"), this simple method may not work. In such a case, we set the sentiment score of the snippet to 0 (neutral).

3.3 User Feedback Integration

As mentioned earlier, a user can interact with the system-generated visual summary and change parts of the summary, e.g., editing the displayed name of an extracted aspect and modifying the sentiment orientation of a snippet. When the user makes a correction, OpinionBlocks does two things: (1) updates its interface for *this user only* to reflect the user input, and (2) sends the user input to the back-end server and stores it in a database. Currently a system administrator decides when to incorporate user feedback to update the system interface for *all* its users.

To incorporate user feedback, OpinionBlocks first selects qualified user changes among all the inputs, and then uses them to update the system. Since users may make mistakes, move things around randomly, or even try to game the system, not all user feedback can be trusted. Similar to adopting crowd-sourced results [5], OpinionBlocks incorporates user feedback only when multiple users report the same error and propose the same solution. It checks the number of identical user-corrections made against a threshold. The threshold is now set by the system administrator and may be different for different user groups (e.g., trustworthy user population versus the general public). For our user studies with the Kindle Fire reviews, we used three as the threshold. That is, if three or more users made the same change, the change is then adopted. For example, the review snippet *"The touch screen has given me no problem so far"* was misclassified as negative by OpinionBlocks. Six participants in our study moved this snippet to the positive row. Thus, OpinionBlocks later marked it as positive.

In practice, user-submitted corrections likely contain conflicts. A very common conflict happens when multiple users identify the same error, but recommend different solutions. For example, the review snippet "*However, hardware volume control, bilateral speakers, and a more thoughtfully placed power button would have earned the Fire 5 stars from me*" was classified as positive by OpinionBlocks. While four participants changed it to neutral, other three moved it to negative. In such cases, OpinionBlocks currently takes the solution by the largest number of "votes", assuming that the number of "votes" passes the threshold described above. Consequently, the sentiment of this review snippet was changed to "neutral". Note that we do not require a "majority rule" here. Our rationale is that when an error is identified by many, it is better to correct it than to leave it in the system, even when there is no consensus on the solution. Adopting the most suggested solution that passes the threshold seems sensible.

4 User Studies

To validate the effectiveness of OpinionBlocks in meeting our two design goals mentioned in the introduction, we conducted user studies to answer two sets of questions:

1. How well does OpinionBlocks support real-world, opinion analysis tasks?
    a) How well can users find important aspects mentioned in the reviews along with their associated sentiment?
    b) How well can users find evidence behind reviewers' opinions?
    c) How well can users get to the detailed facts and discussions as needed?
2. How practical is it for OpinionBlocks to leverage the crowd to improve its quality?
    a) How accurately can users make amendments to correct system errors?
    b) How willing are users to make such contributions?
    c) How well do the amendments improve the system to benefit new users?

4.1 Study Design

To answer the questions mentioned above, we designed two identical studies and conducted them in sequence under two different experimental conditions. Both studies were used to answer the first set of questions and questions 2 (a-b) by steering the participants to identify and correct system analytic errors. In Study 2, however, the user corrections submitted in the first study were incorporated to answer question 2(c). We compared the user performance between the two studies to assess any improvements (e.g., task time) due to user corrections made in the first study. We used disjoint sets of subjects between the two studies, i.e., a between-subject experiment design, to avoid any learning effect.

Participants
Since OpinionBlocks is designed to help end users, we conducted both studies by recruiting participants from Amazon Mechanical Turk (called turkers from now on). After a pilot, we recruited 50 turkers for each study. Turker qualifications included being located in the United States, having done at least 50 approved Human Intelligence Tasks (HITS) on the site, and having over 98% approval rating for all HITS. Each approved task completion was paid $2.59 US dollars. Measures were taken to ensure that one turker could do the task only once.

Data Set
We used Kindle Fire reviews from Amazon.com as our primary data source. We selected this data set for two reasons: First, it is a large data set that can be used to assess user performance in real-world tasks. Second, it is in a domain that may appeal to a general audience. At the time when we conducted the studies, there were over 18,000 reviews on Kindle Fire, with more reviews added daily, indicating people's strong interest in the product. Overall, OpinionBlocks extracted 3034 aspects and 48,000 review snippets from the 18,000 reviews.

Tasks and Measures
Each turker was first directed to an online survey that contained a set of instructions and questions about the tasks. The survey started with a scenario: "*Suppose you want to buy a tablet. You have just heard about Kindle Fire. You'd like to learn more about it so you can make an informed decision.*" The turker was then given a brief tutorial in

a sequence of annotated screen shots of OpinionBlocks, explaining each interface element and function.

After the tutorial, the turker was given a link to launch the *live* OpinionBlocks tool in a separate browser window/tab. After OpinionBlocks was launched, the turker was then instructed to go to the next page of the survey to answer questions using the tool. There were a total of 27 questions in each survey, including fact-finding questions about the product (e.g., "*Which aspect of the product received the most conflicting reviews*") and questions about the tool (e.g., "*How would you rate your experience using our tool to explore the reviews*"). A timer was started when the page of the survey containing the fact-finding tasks was loaded. The timer stopped if all the questions on the page were answered and the page was turned to the next one. The timed duration was used as a measure of completion time for fact-finding tasks.

4.2 Results

We received 50 completed surveys for our first study and 51 for the second one. After reviewing each response, we approved all of them. On average, each turker spent 35.5 minutes on our survey.

1(a) How Well Can Users Identify Important Aspects/Sentiments?
Suppose that users are potential customers in the market for a tablet. We designed two related questions to investigate this aspect. First, we asked them a yes/no question on whether they could make an informed decision on the tablet based on their use of OpinionBlocks. This question was to assess the users' overall confidence in their comprehension of important factors and their associated sentiment to influence their buying decisions. 81 out of the 101 turkers confirmed that the information is sufficient for them to make a decision on the product. One user also provided the rationale for his "Yes" answer: "*there are more green bars than orange*".

The second question asked the turkers to find the important aspects of the product. This question was to examine whether a user's understanding of the aspects was consistent with what the system provided. To do so, we counted the number of times that the users' responses contained at least one of the top-three aspects identified by the system: "*screen*", "*app*", and "*book*". 76 out of the 101 turkers' produced correct answers, indicating user-identified main aspects were consistent with that of the system.

In addition to these two questions, we also used a set of questions such as "*Which aspect has received the most conflicting reviews*?" to assess how well users can use OpinonBlocks to identify aspects with distinct characters (e.g., most positive, negative, and controversial). For these questions, for example, 66% of turkers in Study 1 successfully identified "*screen*" as the aspect that received most conflicting reviews, while 72% turkers did so in Study 2. Moreover, the turkers were able to cite both positive and negative sentiments to substantiate their findings (see more below). Considering that there were 3034 aspects extracted from 18,000 reviews, OpinionBlocks demonstrated its effectiveness in helping users identify salient aspects of the product.

1(b) How Well Can People Find Evidence to Substantiate an Opinion?
We designed three questions to ask the turkers about various review details (e.g., "*What products are the main competitors of the Kindle Fire*?"). For all of these fact-finding questions, users were required to excerpt one or two sentences from the reviews to support their answers. Two coders independently read all turkers' responses (3x101=303 responses from two studies) and marked the responses (Yes or No) based on whether the cited sentences correctly supported the answer. Krippendorff's alpha was computed to measure the inter-coder reliability, where alpha = 0.70, suggesting a good level of consistency between the two coders. We then computed the percentage of turkers that correctly found evidence to back up their answers (0.5 was used when the two coders diverged). Out of 303 responses, 274 were correct (90.4%). Clearly, OpinionBlocks was able to help users find specific evidence for opinions.

1(c) How Well Can People Get to Important Details?
As described above, we learned that users were able to cite relevant evidence to back their answers. However, we also wanted to measure how *accurate* their answers were. To do so, two coders independently read each answer to judge whether it was consistent with the answers suggested by the original data. The inter-coder reliability was measured at alpha = 0.82. The percentage of turkers that gave the correct answer was 95.2%, indicating that the majority of users were able to use OpinionBlocks to find desired details of the product when needed.

2(a) How Accurately Can People Make Amendments?
During the studies, each turker was asked to identify and correct at least ten text analytic errors in OpinionBlocks. Each turker was randomly assigned five aspects displayed in the visual summary to perform this task. From Study 1, we collected a total of 659 user-made changes. Many of the changes were made by multiple participants. After removing the duplicates, we obtained 378 distinct amendments. Among them, 47 corrected misclassification of snippets by aspect; 347 corrected misclassification of snippets by sentiment; and 16 corrected both at the same time. After applying our rules for integrating user feedback, 49 unique amendments were incorporated into OpinionBlocks for Study 2.

Two coders examined the 378 unique changes and coded each of them to assess the correctness of the changes. Due to the inherent semantic ambiguities in interpreting the snippets, the initial independent codings had relatively low inter-coder reliability with alpha=0.36 for both aspect and sentiment placement. This low agreement in perception of aspect-based sentiment is also observed by Brody et al. [6]. Meetings were held between the coders to discuss a more consistent way of coding the results. They identified two common cases of ambiguity and built a set of coding rules: (a) the interpretation of sentiment orientation should be anchored around the aspect first then the product. For example, one snippet stated "*after using the fire for a few weeks now, my ipad is gathering dust.*" If this snippet is under aspect "iPad", then it should be classified as negative, but if under "tablet", it then should be positive; (b) if a change

makes sense or does not make it wrong, count it as correct. For example, the snippet "*Software Controls – I can see why the lack of external buttons would annoy some but for me it is not a problem*" can be interpreted as positive or neutral.

After applying these coding rules, we achieved good inter-coder reliability, 0.91 for aspect and 0.97 for sentiment respectively. The averages of the two coder's ratings were used in the accuracy calculation. For aspect placement, 22 of the 47 changes were coded as correct (46.8%); while 247 of the 347 sentiment changes were accurate (71.2%). These results suggest that people are more capable of fixing sentiment errors than aspect errors. Since the accuracy rates were not as high as we had hoped, we computed the accuracy rate for the 49 changes incorporated by OpinionBlocks, and found that these changes achieved an accuracy of 88.8%. This demonstrates the effectiveness of our user feedback integration rules (section 3.3), and suggests that OpinionBlocks can be improved by crowd-sourced input over the use of state-of-the-art machine learning techniques alone.

2(b) How Willing Are Users to Make Amendments?

We explicitly asked turkers about their willingness to make changes while using the system. From their answers, most users (95%) are willing to contribute.

We also asked the turkers to explain their main reasons for their answers. The reasons given by people who were willing to contribute fell into several categories:

About 50% of the turkers said that they would like to help improve the quality of the tool for its better use. For example, one said, "*I'd be willing to spare a few seconds to improve a tool that I would gladly use.*" Another commented: "*Those features are key to the tool's use" and "it can make the tool more useful and correct*".

About 15% cited the community and social benefits. The reasons include "*I thought this was a useful feature that made the tool more of a community-use tool rather than just an individual-use tool.*"; and "*I think it will go a long way in making users of this app feel like they're contributing in some way. It may even become a draw of sorts for the app.*"

About another 15% felt simply that it was fun and cool to correct things. They mentioned "*It's fun!*", "*It was interesting to correct the errors, because I found myself trying to figure out why each incorrect snippet had been improperly categorized.*"; "*It's cool that you can edit things.*"; "*I like organizing things. Especially when mis-rated reviews stick out like a sore thumb.*"

The majority of people who expressed their unwillingness to contribute (5% of participants) voiced their main concerns about the potential abuse of the system: "*If this was used by multiple people, it would end up being very abused.*"; "*My only concern here is people messing with the system to improve reviews of their own products or make competitors look bad.*"; and "*it's handy but should be checked by someone".*

Other unwilling participants just did not want to bother, or wanted to get paid*: "I'm not really interested in correcting mistakes.*"; and "*I can't see doing it out of the kindness of my heart. If it were on Mechanical Turk I could see doing it for a small amount of money.*"

Overall our results suggest that it is feasible to leverage the power of the crowd to help improve the system.

2(c) How Much Have User-Amendments Made the System Better?
As discussed earlier, the turkers made many changes, of which 49 most common ones were integrated by OpinionBlocks. The incorporated amendments achieved an accuracy of 89%, thus improving the quality of the visual summary.

To measure the impact of integrating the user edits from Study 1 on user tasks, we compared user performance in both studies. To do so, we performed statistical tests using the sequence number of the studies as the independent variable, and all the performance measures taken in the studies as the dependent variables. We found that the turkers' time for completing fact-finding tasks in Study 2 (M=768.1, SD=338.5) was significantly lower than that of Study 1 (M 916.6 seconds, SD =370.2), t_{98}=2.10, p=0.04. Turkers in two studies performed equally well in term of finding correct facts about the products and relevant evidences. In addition, turkers were equally satisfied with our system in both studies. On a 5-point Likert scale, both obtained a median 4 satisfaction ratings, with 5 being "very satisfied".

Overall, our results showed that it is practical to improve the system by leveraging the crowd to correct system errors, and the resulting improved system lets users perform tasks equally well, but significantly faster. One plausible reason for the improved task completion speed is that in the improved system, there is less misplaced unhelpful information, so users do not need to waste time reading.

5 Discussion

Based on our study results, we discuss the limitations and implications of our work.

5.1 Limitations in Text Analytics

OpinionBlocks has adopted several text mining approaches to analyze opinion text and glean useful insights. It also leverages the power of the crowd to help compensate for system mistakes and improve the overall analysis quality. Nonetheless, due to inherent difficulties in text mining, our current approach presents several limitations.

One difficulty is to decide which review snippets to include and how “big” each snippet should be. Currently, OpinionBlocks includes only text snippets following the sentence structure described in Section 3.2. This means it may miss out many useful sentences that do not conform to such a structure. Currently, each snippet contains only one sentence. This might be undesirable in situations where multiple adjacent sentences are used to express an opinion. The challenge is to balance the accuracy and recall when extracting review snippets, as well as balance the size of a snippet to provide sufficient information without overburdening the text analytic engine or the reader. To make the problem more difficult, striking such a balance may depend on factors particular to the data sets.

Another difficulty we have encountered is to determine which aspects to extract. Currently we extract aspects directly out of subject noun phrases, thus covering multiple categories. Besides aspects, such as "screen" and "app", which describe the Kindle Fire, we also extracted "iPad" which is a major competitor of the Kindle Fire. Other extracted aspects, such as "wife", "husband", and "kid", describe possible user groups of the Kindle Fire, and the aspect "problem" falls in a generic category applicable to any product. Depending on users and use cases, some might want to see only the aspects pertinent to the product, while others may want to learn more about the aspects of competing products (e.g., aspects of iPad in the context of Kindle). More work is needed to make aspect extraction more meaningful and extensible.

5.2 Common Ground versus Personalization

In the User Studies section, we show that opinions are often ambiguous and that different people may interpret them very differently. Building a "ground-truth" of opinion summary is non-trivial and is unlikely to satisfy every user. Allowing a certain degree of personalization may be desirable in support of individual users' decision making. Currently, OpinionBlocks allows each user to make amendments that affect only that user's private session. These changes are propagated more widely when the system administrator decides to do so, and only high-quality changes suggested by many users are adopted. Thus, the standard version of OpinionBlocks that every user starts with is quality controlled, even though users may make amendments to their own private sessions. Complications may arise when merging divergent sets of amendments from many users. This will certainly make a good future research topic.

5.3 Potential System Abuse

A few participants of our user studies expressed their concerns over potential abuse of a system like OpinionBlocks, including trolling or businesses manipulating the information for their own commercial gains through user amendments of opinion summaries. Currently, OpinionBlocks gives the system administrator a great deal of control over which amendments can be integrated into the system. The system administrator can decide to tighten or loosen the threshold for integration or filter out changes from certain users. While further research is required to figure out how to best monitor and moderate user behavior, one approach is to the leverage the crowd themselves. As shown in our user studies, the accuracy of aggregated user amendments is much higher than that of individual changes. This means aggregation of crowd input may help prevent or reduce malicious behavior. Currently our aggregation rules are very simple, future research is needed to develop more sophisticated rules, e.g., incorporating information such as the degree of difficulty of text analytic tasks and user reputation.

5.4 Fostering Healthy Online Review Communities

Our work bears a major implication on the research in online communities. Gilbert and Karahalios [11] pointed out two problems of current review sites: (1) large numbers of reviews are never read and in essence wasted; and (2) "pro" reviewers dominate the community and it's hard to hear the voice of "amateur" reviewers. They call on system designers to nudge community members toward community-wide goals. OpinionBlocks helps address both problems: It summarizes the reviews and helps users understand large collections of reviews. It also fosters a democratic environment for others to contribute. In short, we have taken the first step to create a platform to foster a healthier online community where users can potentially help the system and help one another.

5.5 Value to Text Analytics Research

It is also worth noting that our approach of marrying machine and human intelligence to text analytics produces invaluable assets for text analytics research. First, crowd feedback can be used as an indicator to identify "high-value" areas for users. As shown by our study results, users mostly made corrections to the sentiment classification but only a few on the aspect classification. This suggests that users may be more sensitive to certain types of errors than others. Moreover, user-submitted corrections can be used as a training corpus to help tune analytic algorithms.

6 Conclusion

We have presented OpinionBlocks, a novel visual analytic system that aids users to analyze large sets of opinion text. It is uniquely designed to combine state-of-art NLP technologies with crowdsourcing to aid users in their real-world opinion analysis tasks. It employs multiple NLP technologies to automatically generate a fine-grained, aspect-based visual summary of opinions. As demonstrated by our user studies involving 101 users on Amazon Mechanical Turk, the majority of participants not only were able to use OpinionBlocks to complete real-world opinion analysis tasks, but they also exhibited a surprisingly high degree of altruism and concerns for the well-being of online review communities. As users gain value from the system, they become willing contributors to help correct system analytic errors and improve the system. Moreover, the crowd-assisted system enhancement significantly improved task completion time. Based on these findings, combining visual analytics with crowd-sourced correction is thus shown both feasible and effective.

References

1. Sentiment lexicon, `http://www.cs.uic.edu/~liub/FBS/opinion-lexicon-English.rar` (accessed: September 09, 2012)
2. Stanford log-linear part-of-speech tagger, `http://nlp.stanford.edu/software/tagger.shtml` (accessed: September 09, 2012)

3. Amershi, S., Fogarty, J., Weld, D.: Regroup: interactive machine learning for on-demand group creation in social networks. In: CHI, pp. 21–30 (2012)
4. Amershi, S., Lee, B., Kapoor, A., Mahajan, R., Christian, B.: Cuet: human-guided fast and accurate network alarm triage. In: CHI, pp. 157–166 (2011)
5. Bernstein, M.S., Little, G., Miller, R.C., Hartmann, B., Ackerman, M.S., Karger, D.R., Crowell, D., Panovich, K.: Soylent: a word processor with a crowd inside. In: Proceedings of the 23nd Annual ACM Symposium on User Interface Software and Technology, UIST 2010, pp. 313–322. ACM, New York (2010)
6. Brody, S., Elhadad, E.: An unsupervised aspect-sentiment model for online reviews. In: Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics, HLT 2010 (2010)
7. Carenini, G., Rizoli, L.: A multimedia interface for facilitating comparisons of opinions. In: IUI, pp. 325–334 (2012)
8. Carlier, A., Ravindra, G., Charvillat, V., W., O.: Combining content-based analysis and crowdsourcing to improve user interaction with zoomable video. In: ACM MM, pp. 43–52 (2011)
9. Faridani, S., Bitton, E., Ryokai, K., Goldberg, K.: Opinion space: a scalable tool for browsing online comments. In: CHI 2010, pp. 1175–1184. ACM, New York (2010)
10. Fieller, E., Hartley, H., Pearson, E.: Tests for rank correlation coefficients. I. Biometrika 44(3/4), 470–481 (1957)
11. Gilbert, E., Karahalios, K.: Understanding deja reviewers. In: Proceedings of the 2010 ACM Conference on Computer Supported Cooperative Work, CSCW 2010, pp. 225–228. ACM, New York (2010)
12. Hu, M., Liu, B.: Mining and summarizing customer reviews. In: KDD, pp. 168–177 (2004)
13. Huang, J., Etzioni, O., Zettlemoyer, L., Clark, K., Lee, C.: Revminer: An extractive interface for navigating reviews on a smartphone. In: UIST (2012)
14. Jo, Y., Oh, A.: Aspect and sentiment unification model for online review analysis. In: Proceedings of ACM Conference on Web Search and Data Mining, WSDM 2011 (2011)
15. Lee, Y.E., Benbasat, I.: Interaction design for mobile product recommendation agents: Supporting users' decisions in retail stores. ACM Trans. Comput.-Hum. Interact. 17(4), 17:1–17:32 (2010)
16. Liu, B., Hu, M., Cheng, J.: Opinion observer: analyzing and comparing opinions on the web. In: WWW, pp. 342–351 (2005)
17. Liu, B., Zhang, L.: A Survey of Opinion Mining and Sentiment Analysis. In: Mining Text Data. Kluwer Academic Publishers (2012)
18. Liu, B.: Sentiment analysis and opinion mining. In: Synthesis Lectures on Human Language Technologies. Morgan & Claypool Publishers (2012)
19. Moghaddam, S., Ester, M.: ILDA: interdependent LDA model for learning latent aspects and their ratings from online product reviews. In: Proceedings of the Annual ACM SIGIR International Conference on Research and Development in Information Retrieval, SIGIR 2011 (2011)
20. Mukherjee, A., Bing, L.: Aspect Extraction through Semi-Supervised Modeling. In: Proceedings of 50th Anunal Meeting of Association for Computational Linguistics, ACL 2012 (2012)
21. Nelson, P.: Information and consumer behavior. J. of Political Economy 78(2), 311–329 (1970)
22. OpenNLP, `http://opennlp.apache.org`
23. Pang, B., Lee, L.: Opinion mining and sentiment analysis. Foundations and Trends in Information Retrieval (2008)

24. Park, D., Lee, J., Han, J.: The effect of online consumer reviews on consumer purchasing intention: The moderating role of involvement. Intl. J. of E-Commerce 11(4), 125–148 (2007)
25. Patel, K.: Lowering the barrier to applying machine learning. In: Adjunct Proceedings of the 23nd Annual ACM Symposium on User Interface Software and Technology, UIST 2010, pp. 355–358. ACM, New York (2010)
26. Rohrdantz, C., Hao, M.C., Dayal, U., Haug, L.-E., Keim, D.A.: Feature-based visual sentiment analysis of text document streams. ACM Trans. Intell. Syst. Technol. 3(2), 26:1–26:25 (2012)
27. Schuler, K.K.: VerbNet: A Broad-Coverage, Comprehensive Verb Lexicon. PhD thesis, University of Pennsylvania (2006)
28. Snow, R., O'Connor, B., Jurafsky, D., Ng, A.: Cheap and fast—but is it good?: evaluating non-expertannotations for natural language task. In: Proc. EMNLP 2008, pp. 254–263 (2008)
29. Thet, T., Na, J., Khoo, C.: Aspect-based sentiment analysis of movie reviews on discussion boards. J. Inf. Sci. 36(6), 823–848 (2010)
30. Wu, Y., Wei, F., Liu, S., Au, N., Cui, W., Zhou, H., Qu, H.: Opinionseer: Interactive visualization of hotel customer feedback. IEEE Trans. Vis. Comput. Graph. 16(6), 1109–1118 (2010)
31. Yatani, K., Novati, M., Trusty, A., Truong, K.: Review Spotlight: a user interface for summarizing user-generated reviews using adjective-noun word pairs. In: CHI, pp. 1541–1550 (2011)
32. Zhu, F., Zhang, X.: Impract of online consumer reviews on sales: The moderating role of products and consumer. J. of Marketing 74, 133–148 (2010)