Leveraging the Crowd to Improve Feature-Sentiment Analysis of User Reviews

Shih-Wen Huang[1], **Pei-Fen Tu**[1], **Wai-Tat Fu**[1], **Mohammad Amanzadeh**[2]
Department of Computer Science[1], Industrial & Enterprise System Engineering[2]
University of Illinois at Urbana-Champaign
{shuang51, ptu3, wfu, amanzad2}@illinois.edu

ABSTRACT

Crowdsourcing and machine learning are both useful techniques for solving difficult problems (e.g., computer vision and natural language processing). In this paper, we propose a novel method that harnesses and combines the strength of these two techniques to better analyze the features and the sentiments toward them in user reviews. To strike a good balance between reducing information overload and providing the original context expressed by review writers, the proposed system (1) allows users to interactively rank the entities based on feature-rating, (2) automatically highlights sentences that are related to relevant features, and (3) utilizes implicit crowdsourcing by encouraging users to provide correct labels of their own reviews to improve the feature-sentiment classifier. The proposed system not only helps users to save time and effort to digest the often massive amount of user reviews, but also provides real-time suggestions on relevant features and ratings as users generate their own reviews. Results from a simulation experiment show that leveraging on the crowd can significantly improve the feature-sentiment analysis of user reviews. Furthermore, results from a user study show that the proposed interface was preferred by more participants than interfaces that use traditional noun-adjective pair summarization, as the current interface allows users to view feature-related information in the original context.

Author Keywords

Human computation; crowdsourcing; interactive machine learning; sentiment analysis; user generated content

ACM Classification Keywords

H.5.2 [Information interfaces and presentation]: User Interfaces.

INTRODUCTION

With the rapid success of Web 2.0 technologies, user-generated content has become a major source of online information. One of the most notable examples is online reviews. Millions of people now write reviews on websites like Yelp or Amazon to express their opinions regarding different restaurants or products. These user-generated reviews can be very helpful for others to make wiser decisions.

However, extensive amounts of user-generated reviews are difficult for people to digest, creating a typical problem of information overload. There are two potentially conflicting goals when designing systems that leverage on user-generated reviews. First, the system should mitigate information overload by summarizing important information for the users; second, the system should allow users to express their experiences in their own words while at the same time allowing others to read their reviews in context. With the proposed system, we aim to provide a balance between the two.

Summarizing the information in user reviews based on related features and sentiment has proven to be an effective way to help users to digest the massive amount of information more efficiently. For example, Review Spotlight [33] performs feature-sentiment analysis and presents the results using noun-adjective pairs. Yatani *et al.* [33] showed that this interface can help users make decisions significantly faster, which demonstrates that feature-sentiment information can help users digest user-generated reviews more efficiently.

The feature-sentiment analysis in existing intelligent interfaces [13, 18, 33] typically incorporates two steps. First, it finds the features by identifying nouns with high frequency counts in the text. Second, after determining the features, it uses the adjective near each feature and a predefined glossary (e.g., SentiWordNet [9]) to determine the feature's orientation. However, researchers have pointed out that this unsupervised learning approach has two major disadvantages, which we will summarize below.

First, as mentioned in [12], this kind of analysis typically can discover only features that are explicitly discussed in the content. For example, consider the following sentence:

"While light, it will not easily fit in pockets."

This sentence is related to size of a product; however, it is very difficult for this unsupervised learning approach to discover the feature because the word "size" does not appear in the sentence [12]. This greatly undermines the utility of the algorithm since many opinions expressed in user-generated reviews are implicit, and they tend to elude discovery by unsupervised learning methods.

IUI'13, March 1922, 2013, Santa Monica, CA, USA

The second disadvantage is that this approach often makes mistakes in selecting useful features and deciding the sentiment orientation of the features. For instance, as mentioned in [33], "impeccable", which should be a positive word, has a very high negative score in their system. Illustrating similar mistakes made by the system, some of the system's users noted that the features they presented did not make much sense (as described in the paper [33]). These errors may lower the motivation to use a system as users perceive it to be unreliable.

One way to address these problems is to use labeled data and supervised learning to find the hidden features in the reviews. Supervised learning has demonstrated better performance than unsupervised methods in general [3]. Moreover, this approach can identify sentences related to a feature even if the feature itself is not in the sentence because it uses more than a single term in the sentences to classify. However, supervised methods are difficult, if not impossible, to implement in existing interfaces because it is difficult to motivate users to create a label for each sentence they write. Therefore, one big challenge for this approach is collecting labeled data. Finding a way to motivate users to provide labels is key to the success of this supervised approach.

To address the questions discussed above, *we propose a novel intelligent interface that collects training data directly from users as they generate reviews – a concept often called implicit crowdsourcing.* The system we built can perform feature-sentiment analysis in nearly real-time. As a result, it can provide predictions while a user is writing the review. If the user sees the prediction is wrong, the user can simply click the icons on the interface to correct the erroneous prediction. Therefore, instead of providing labels for every sentence in the review, users need only to correct some mistakes made by the system, which greatly reduces the effort necessary to provide feature-sentiment information. In addition, users are more motivated to provide labels because these labels are related to the accuracy of their own reviews. The collected data can be used as new training instances for the classifier. Moreover, increasing the number of training instances raises the coverage of the supervised learning algorithms [14]. Therefore, leveraging the crowd to collect user-generated labels allows the classifiers to provide more accurate predictions as the number of users in the system grows. To preview our result, the experiment shows that our supervised classifiers can achieve much higher F_1 scores than baseline models that discover feature-sentiment information using unsupervised methods.

Another drawback of many existing intelligent interfaces is that the feature-related information is summarized in a very compressed form (e.g., noun-adjective pairs). Although this has the advantage of allowing users to retrieve the feature-related information in the reviews more quickly, it also destroys the original context of the reviews, which often contain more than pure information, such as social cues, personal expression, etc. In contrast, the proposed system provides a highlighting function that highlights the feature-related information in its original context. This function integrates into the traditional review reading experience and creates a good balance between focusing on feature-related information and understanding the full reviews. In this study, we compared the two designs (i.e., noun-adjective pair summarization and highlighting) to observe what users liked or disliked about these two types of systems. To preview our results, we did find that most users preferred to see the original context rather than the purely summarized features, thus providing support to the design.

Overview of the paper

In the rest of the paper, we first will review related work on how crowdsourcing can be used to assist supervised learning and feature-sentiment analysis of user reviews. We will then describe the current system and how it differs from previous ones. Then we will perform two sets of evaluation. First, we will present results from a simulation experiment to demonstrate how the system can outperform previous systems that utilize unsupervised learning, and how the system can improve over time as more user-generated labels are collected to improve the classifier in the current system. Second, we will present results from a user study that tested whether review readers and writers would like the features of the current system. Specifically, we tested the extent to which review readers would like to see the context of user reviews instead of merely summarized reviews, and we tested whether the real-time feedback would encourage review writers to provide labels for their own reviews. Finally, we will discuss the implication of our results for the design of systems that rely on user-generated content in general, as well as the future direction of the current research.

RELATED WORK

Enhancing machine learning algorithms by collecting labeled data from crowdsourcing

Crowdsourcing has been proven as an effective way to solve various kinds of problems [11]. Individuals can easily recruit online workers from crowdsourcing platforms like Amazon Mechanical Turk (AMT)[1] to solve problems that are difficult for digital computers (e.g., text editing [1] and answering visual query [2]) at a very low cost. These examples have merely begun to demonstrate the potential of crowdsourcing as a social computing technique that can be applied in a wide range of situations.

One of the most notable application for crowdsourcing is collecting labeled data to improve the performance of machine learning algorithms. Von Ahn and Dabbish [29] pioneered the field by developing the ESP game, which recruits people to generate image labels while playing an online game. By 2008, this game had recruited 200,000 players and collected more than 50 million labels [30]. The collected labels then were used to improve Google image search. Other games [10, 31] and methods [22, 25, 26] also have been proposed to collect high-quality image labels to train computer vision algorithms.

[1] https://www.mturk.com

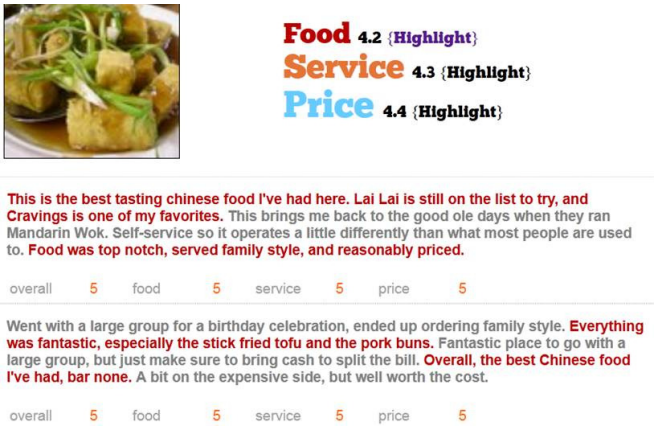

Figure 1. When reading reviews, users can highlight sentences that are related to the aspect which interests them by a single click.

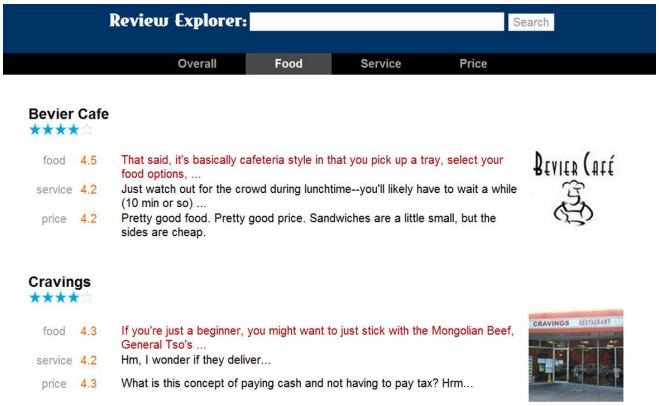

Figure 2. When browsing entities, the system allows users to sort them (e.g. restaurants) using their ratings of different features.

In addition to image labels, crowdsourcing also has been used to collect training data for natural language processing. Snow *et al.* [24] studied how the labeled data generated by the non-experts from AMT can be more cost-effective than those generated by experts. They also showed that the labels collected from crowdsourcing could successfully improve machine learning algorithms. There are also various workshops in NAACL [4], SIGIR [17], and WSDM [16] that aim at utilizing crowdsourcing to generate labeled data that are useful to data mining and information retrieval.

Instead of explicitly recruiting crowd workers to generate labeled data to enhance machine learning, Nichols *et al.* [19] proposed *implicit crowdsourcing*, a method that directly collects data generate by the users, which is different from traditional crowdsourcing that pays money to recruit workers from AMT to create labeled data. This allows the system to collect more data as the number of users increases. For example, they [19] found that by collecting status updates posted to Twitter, the system can successfully generate meaningful summaries of sporting events. This design is especially useful for supervised learning because the size of training data is essential to the performance of the algorithms [14]. Besides summarizing the data that have been already posted, the intelligent interface we built can generate the predictions in real time and involve users to correct the errors made by the system. As a result, our interface further uses artificial intelligence to assist users to generate labeled data more easily. To the best of our knowledge, this is a novel concept that has not yet been explored.

Intelligent interfaces for user reviews

Since user reviews contain much valuable information, many researches have proposed different methods to analyze the features and sentiments expressed in user reviews [20]. Hu and Liu [12] used the minimum support of association rule to identify frequent terms and phrases in user reviews as features. Many researchers [7, 21, 28, 32] also studied how to use machine learning algorithms to classify the sentiments expressed in user-generated reviews.

Recently, many intelligent interfaces have been developed to help users to understand feature-sentiment information in a huge amount of user reviews. Liu *et al.* [18] implemented Opinion Observer, an interface that uses bar charts to present the positive and negative sentiments of each feature. Carenini *et al.* [5] constructed a treemaps interface for users to interactively explore the information that interests them. In addition, they also designed a novel visualization for users to compare the feature-sentiment information between different entities [6]. Yatani *et al.* [33] developed Review Spotlight to present feature-sentiment information in user reviews using noun-adjective pairs in a tag cloud. Huang *et al.* [13] further group the similar features together to display feature-sentiment information in a more concise format. The biggest difference between the intelligent interface of our system and the existing ones is that: instead of summarizing the information using noun-adjective pairs, our system presents the information by highlighting the feature-related sentences in their original context. This allows the users to focus on feature-related information while still have the opportunity to explore other information in the reviews.

Moreover, intelligent interfaces also have been used to assist review writing. Dong *et al.* [8] developed Reviewer's Assistant, a browser plug-in that identifies the sentences written by previous users which might also be used by the current user and recommend them to the user. Their study showed that the system could suggest sentences that were actually written by the users. The current system also has an intelligent interface for review writing. Nevertheless, we aim at collecting feature-sentiment information that can be helpful to the readers instead of assisting reviewers to generate user reviews.

SYSTEM DESIGN AND IMPLEMENTATION

The proposed system incorporates two functions that help readers digest user reviews: first, it allows its users to highlight sentences based on the features they are interested in by a single click; this greatly reduces the amount of information that a user must read. (Figure 1) Second, users of the system can rank the entities based on their feature ratings, which are inferred directly from the contents. (Figure 2)

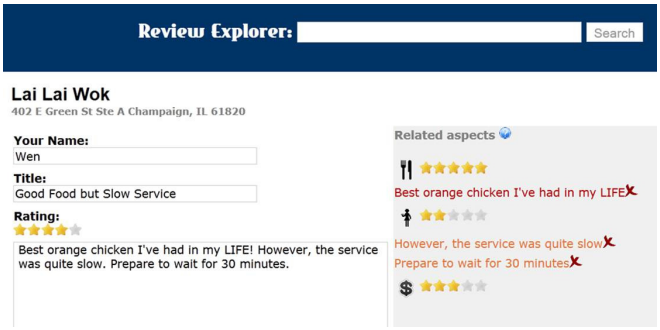

Figure 3. Users can click the icons on the interface to correct the erroneous predictions made by the system while composing reviews. For example, the cross sign near the sentences allow users to cancel the predictions made by the system. In addition, users also can click the stars to change feature-ratings.

To collect the information needed to perform the two functions mentioned above, the system also has a novel intelligent interface that conducts feature-sentiment analysis in real time. When a user is writing a review using the system, whenever the user finishes a sentence, the system would provide the user real-time predictions about the feature(s) that are related to it and the star ratings of the features. If the user feels that the predictions made by the system are wrong, they can simply click the icons on the interface to correct the errors. (Figure 3) A graphical representation of the design of our system is shown in Figure 4.

Data and features of the current system

The data used in the current system was retrieved from Yelp's Academic Dataset[2], which consists of 87,173 reviews of restaurants near 30 schools. In this study, we used three pre-defined features: food, service, and price. However, one should notice that the data and features of the system easily can be altered or expanded and are not limited to the current settings.

Supervised two-layer feature-sentiment analysis

To discover the related features of each sentence and the inferred ratings of the features, a supervised two-layer feature-sentiment analysis was conducted on each review in the corpus. A graphical representation of the flow of the two-layer analysis is shown in Figure 5.

The first layer of the analysis is the *sentence feature classification*. In this layer, the system decides if one sentence is related to a target feature or not. To initiate the classifiers of the system, we collected 5,000 labeled sentences by recruiting 194 workers from Amazon Mechanical Turk at a cost of $9.70 (from 4/11/2012 to 4/21/2012). The workers were asked to label whether a sentence was related to any of the three pre-defined features or to none of them (A sentence can be related to more than one feature). This is used to simulate the data generated by the initial users of our system. We preprocessed the text by stemming and removing stop words, and we converted these labeled sentences into unigram feature vectors.

[2]http://www.yelp.com/academic_dataset

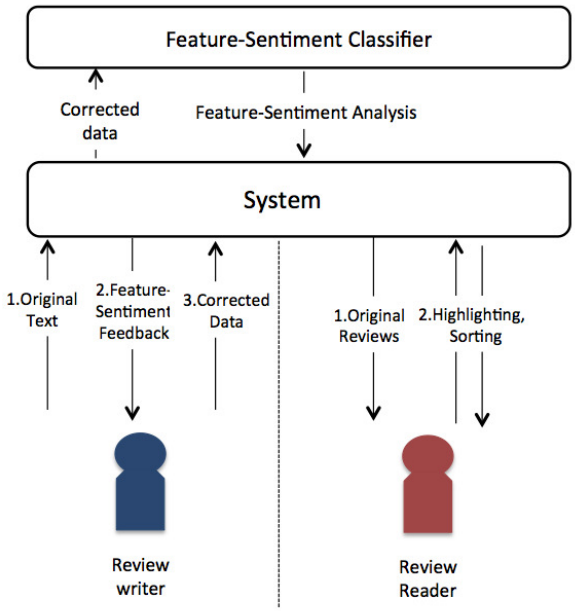

Figure 4. When user is writing a review using our novel intelligent interface, the system performs a real-time feature-sentiment analysis. The user can easily correct the erroneous predictions made by the system. The corrected data then is used to improve the classifiers to provide useful information for other users to digest the reviews.

Then, we used the SVM^{light} package[3] to train the classifiers on these feature vectors. If a sentence was related to a certain feature, its associated feature vector would be treated as a positive training instance for the classifier of that feature. In contrast, if the sentence is not related to that feature, its feature vector was used as a negative training instance. Finally, sentences that did not have labels in the corpus were converted into unigram feature vectors, and the system used the classifiers trained on the labeled sentences to classify the features related to the unlabeled sentences.

The second layer of the analysis is the *feature-rating prediction*, which predicts the star ratings of different features in each review. To construct the classifiers, we utilized the reviews and their associated star ratings in Yelp's academic dataset. First, the reviews with 4 or more stars were used as positive training instances and reviews with less than 4 stars were used as negative training instances. These reviews were converted to feature vectors and were used to train a positive-negative classifier using SVM^{light}. By a similar procedure, we built a 4-5-star classifier and a 2-3-star classifier[4]. These classifiers allowed us to predict the overall ratings of each review. For example, if a review was predicted as positive (more than 3 stars) by the positive-negative classifier, the system then would run the 4-5-star classifier to see if it should be classified as a 4-star review or a 5-star review.

Equipped with the rating classifiers, the system then predicts

[3]http://svmlight.joachims.org/

[4]We grouped 1-star reviews to the 2-star reviews category because there is only a very small portion of the reviews are 1-star

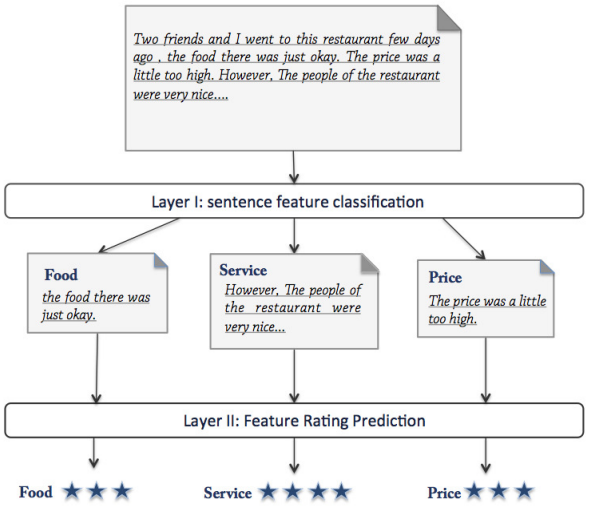

Figure 5. The flow of the two-layer feature-sentiment analysis. The system first classify the sentences with their related features. The sentences that are related to the same feature then are grouped together and are used to predict the rating of the feature.

the feature ratings of each review by classifying the star rating using only the contents that are related to a feature (based on the first layer analysis). For instance, when predicting the service rating of a review, the system would first find out which sentences related to service using the service-feature classifier. Then, these sentences would be grouped together and classified by the star-rating classifiers. Finally, the output of the classifier would be the service rating of the review.

Collecting corrected feature-sentiment information from users

To ensure the accuracy of the analysis made by the system and collect more training data to improve the performance of the classifier, the system provides an interface that can perform real-time feature-sentiment analysis while a user composes a review. Whenever the user finishes a sentence, the web-based system sends it to the server using AJAX. When the server receives the sentence, a Python script performs the text preprocessing and converts the sentence into a unigram feature vector that can be processed by the classifiers. Then the system conducts the two-layer feature-sentiment analysis using the SVM^{light} package. Finally, the result of the analysis is sent back to the interface on the client side. The whole analysis can be performed within one second, including the latency of the Internet, so for the user, the analysis seems to occur in real time.

If the predictions are wrong, the user can simply click the icons next to the predictions on the interface to correct them. For example, if a sentence is mistakenly classified as a sentence that is related to price but the user judges that it is not, the user can click the cross sign near the sentence to cancel this prediction and assign the sentence to other categories, or not to any existing category. If, for example, the sentence is judged to be related to food, the user can click the icon that represents food near the sentence to label it. Furthermore, if the star-rating predictions are wrong, the user can click the stars on the interface to change the ratings.

EXPERIMENTS

To evaluate whether our proposed method really improves feature-sentiment analysis, we manually labeled the related features (i.e., food, service, and price) of 1,000 sentences in the dataset and used these as gold standard test dataset. The system's ability to discover the related features of our proposed design and the baseline models was evaluated on this dataset.[5] In this study, we focused only on the ability to classify feature-related sentences of the proposed method and left evaluation of the ability to generate accurate feature-rating for future work. Specifically, we wanted to test two hypotheses:

H1: The supervised classifier in our design can achieve higher performance than a traditional unsupervised approach can.

H2: More training data improves the performance of the classifier.

Two experiments were performed to test the hypotheses. The details of these experiments are described below.

Experiment I: Comparisons between supervised and unsupervised methods

In this experiment, we compared the proposed supervised method to a baseline model with unsupervised learning to test if **H1** is true.

Method

Three sentence-feature classifiers (food, service, and price) were trained on 5,000 sentences labeled by AMT workers. We performed text preprocessing which includes stemming and removing the stop words of the sentences. The sentences then were converted to the feature vectors using a unigram model. Finally, we used SVM^{light} to train the classifiers on these feature vectors.

In addition, we built a baseline model similar to the ones in [13, 18, 33]. This unsupervised model used 434,664 sentences in the full data set. The data size is much larger than the 5,000 labeled sentences used in the supervised method. To identify the frequent features in the sentences, we first used the part-of-speech tagging function in NLTK[6] to find all the nouns and adjectives in the sentences. Then, we performed the same text preprocessing as in the supervised method. After that, the nouns (after stemming) that appeared in more than 1% of the total sentences were selected as the features. This threshold is the same as the minimum support used in [18]. We tried to vary the threshold to 0.1%, 0.5%, and 2%, but there were no significant differences in the results of the various thresholds. Therefore, only the results of the 1% threshold were reported. After the features were selected, the closest adjective to each feature was considered as the one

[5]We did not use the labels retrieved from AMT to evaluate the classifiers because we found that it contains many low-quality labels.
[6]http://nltk.org/

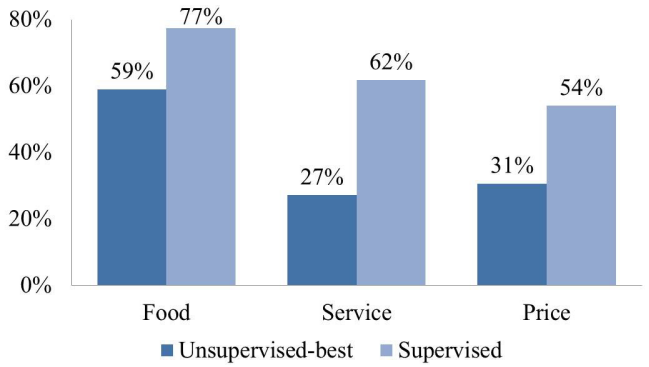

Figure 6. The comparisons between the F_1 scores of the supervised method and the best performance of the unsupervised methods. The results show that the supervised method can achieve much better F_1 scores than the unsupervised methods in all three features tested.

that described the feature. We then grouped the features using the Kullback-Leibler divergence [15] between the adjectives that described the features, which is the feature grouping method used in [13]. Finally, the top (5, 10, 20, 30) closest features to food, service, and price were assigned to them as sub-features. If one of the sub-features occurred in a target sentence, the sentence was classified as related to the main feature (food, service, or price).

Evaluation

We evaluated the systems on a 1000-sentence test data manually labeled by the authors. The precision, recall, and F_1 score were calculated using the formulas below:

$$\text{precision} = \frac{\text{\# feature-related sentences classfied correctly}}{\text{\# sentences classified as related to the feature}}$$

$$\text{recall} = \frac{\text{\# feature-related sentences classfied correctly}}{\text{\# feature-related sentences}}$$

$$F_1 = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision + recall}}$$

We used the F_1 scores to evaluate the performance of the classifiers because it is a weighted average of precision and recall. Since there is a trade-off between precision and recall, F_1 scores can evaluate the results more fairly [23]. The precision, recall, and F_1 scores of the supervised method and the unsupervised methods with various number of sub-features are summarized in the table 1, 2, and 3.

The results show that the proposed supervised method achieved much higher F_1 scores in classifying feature-related sentences. (Figure 6) When looking more carefully into the precision and recall of each classifier, we see that the supervised method can achieve much higher precision than the unsupervised methods can. In addition, although increasing features to the unsupervised methods allows them to outperform the supervised method in recall of two of the three features,

	5 feat.	10 feat.	20 feat.	30 feat.	supervised
Food	47.67%	51.69%	53.78%	51.99%	85.75%
Service	18.47%	18.39%	16.59%	17.09%	90.00%
Price	24.55%	16.29%	13.83%	11.91%	80.95%

Table 1. Precision of supervised method and unsupervised methods with different number of sub-features

	5 feat.	10 feat.	20 feat.	30 feat.	supervised
Food	30.09%	41.63%	61.09%	68.10%	70.50%
Service	34.33%	41.04%	50.75%	65.67%	47.01%
Price	31.68%	35.64%	49.50%	63.28%	40.50%

Table 2. Recall of supervised method and unsupervised methods with different number of sub-features

	5 feat.	10 feat.	20 feat.	30 feat.	supervised
food	36.89%	46.12%	57.20%	58.96%	77.38%
Service	24.02%	25.40%	25.01%	27.12%	61.76%
Price	30.60%	24.51%	22.22%	20.00%	53.99%

Table 3. F_1 scores of supervised method and unsupervised methods with different number of sub-features

the precision becomes unacceptably low (around 15%). The reason is that unsupervised methods would include many general terms as sub-features, which are not very helpful to the classification task. In contrast, the supervised method uses the unigram feature vector to determine whether one sentence is related to a feature. Therefore, the classification result is not dominated by any single term. In addition, the supervised method can discover some hidden patterns in sentences. For instance, consider the following:

"The food is superb and it comes out pretty fast."

This sentence is related to both food and service. By using an unsupervised method, the only feature that can be discovered is food because it is mentioned explicitly in the sentence. On the other hand, the supervised method used in our system can successfully capture both features because "fast" and "come" both carry meanings that are related to service, which is the hidden feature of this sentence. The supervised method can learn these implicit concepts (e.g., fast and come) to discover the hidden feature if it is trained on a massive amount of labeled data. As a result, the supervised method can achieve higher F_1 scores. The results therefore provide support to **H1** – using supervised learning to train the classifier can result in better performance than that achieved by traditional unsupervised methods. In particular, the current method can significantly improve precision because of the fact that many hidden variables that define the categories in user reviews cannot be easily identified by unsupervised methods.

Experiment II: Comparisons between the supervised methods with various training data size

In this second experiment, we varied the size of data that is used to train the supervised classifiers to see if **H2** is supported.

Method

We trained the classifiers on 0.5K, 1K, 2K, 3K, 4K, and 5K labeled sentences. These subsets of labeled sentences were selected randomly from the 5,000 labeled sentences collected from AMT as described in experiment I.

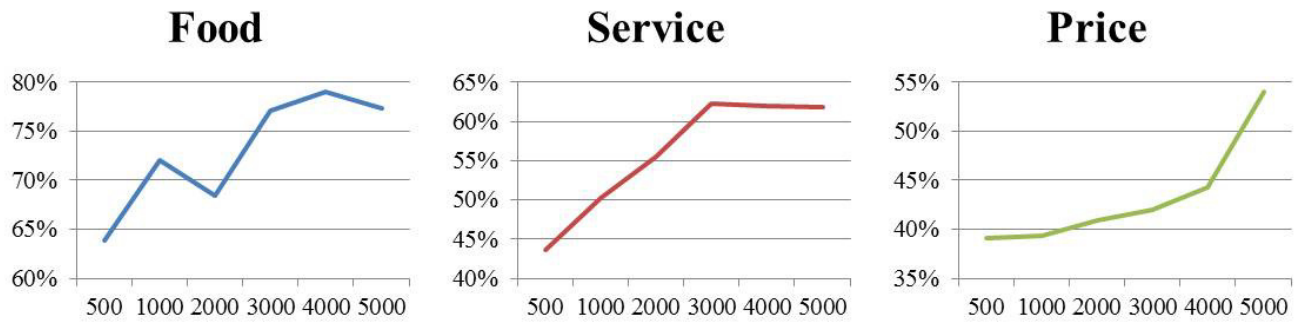

Figure 7. The F_1 scores of the feature-classifiers trained on various size of data. The results show that there was a very high positive correlation between the size of training data and the performance of the classifiers.

Evaluation

The precision, recall and F_1 scores of the feature-classifiers trained on various sizes of labeled sentences are summarized in the table 4, 5, and 6.

	500	1000	2000	3000	4000	5000
food	85.02%	82.13%	81.37%	82.90%	84.58%	85.75%
Service	95.00%	68.83%	70.93%	86.67%	85.53%	90.00%
Price	92.59%	96.15%	87.10%	78.38%	79.49%	80.95%

Table 4. Precision of supervised method on various size of training data

	500	1000	2000	3000	4000	5000
food	51.13%	64.19%	59.01%	72.07%	74.10%	70.50%
Service	28.36%	39.55%	45.52%	48.51%	48.51%	47.01%
Price	24.75%	24.75%	26.73%	28.71%	30.69%	40.50%

Table 5. Recall of supervised method on various size of training data

	500	1000	2000	3000	4000	5000
food	63.86%	72.06%	68.41%	77.11%	78.99%	77.38%
Service	43.68%	50.23%	55.45%	62.20%	61.91%	61.76%
Price	39.06%	39.37%	40.91%	42.03%	44.28%	53.99%

Table 6. F_1 score of supervised method on various size of training data

The results show that the performances of the classifiers clearly are positively correlated to the size of training data. The Pearson correlations between the size of training data and the F_1 scores of the feature classifiers for food, service, and price are 85%, 90%, and 89%, respectively. We also found that the improvements of the classifiers are mostly from recall. The Pearson correlations between the size of training data and the recall of the feature classifiers for food, service, and price are 81%, 79%, and 91% respectively. This shows that more training data can help the classifiers discover more hidden patterns in the data, which supports **H2**, that more training instances can enhance the performance of the feature classifiers. This also implies that when more users use the system, the classifiers behind the system can generate more accurate predictions because more training data can be collected from users.

Summary

To summarize, our experiments show that the proposed supervised method can be a better mechanism in classifying sentences by their related features than traditional unsupervised method. Moreover, increasing the training data can further improve the performance of the supervised classifiers, which means that collecting labeled data from users effectively enhances the feature-sentiment analysis of the system.

USER STUDY

We conducted a user study to gain more insight into how our interface impacts users and to answer the following three questions that are important to our theses:

1. Does the feature-sentiment analysis and highlighting function of our system help users digest the information in user reviews?
2. Is the highlighting function better than existing intelligent interfaces that summarize feature-sentiment by noun-adjective pairs?
3. Will users correct the erroneous predictions made by the system?

The first two questions show how our proposed interface helps readers digest user reviews and the last question lets us know if the system really can collect more corrected feature-sentiment information when the system is deployed in the wild.

Procedure

At the beginning of the study, the participants were asked to fill out a questionnaire collecting demographic information and asking how often they read and write user reviews. Then, we introduced our interface and its functions to the participant. After that, the participants were asked to read reviews about a restaurant using both our interface and Yelp's website[7]. When they finished reading, they were asked if the highlighting function of our interface helped them get useful information about the restaurant from the reviews.

Then, we introduced RevMiner[8] [13], an interface that uses noun-adjective pairs to summarize feature-sentiment information in the reviews to its users. (Figure 8) After the participants became familiar with the interfaces, we let them compare the noun-adjective pair summarization with our highlighting function. We chose to use the RevMiner interface

[7]http://www.yelp.com/
[8]http://revminer.com/

Figure 8. Noun-adjective pairs visualization in RevMiner. We used RevMiner as an example to represent the interfaces that summarize feature-sentiment information using noun-adjective pairs.

because it was one of the most recently developed at the time this article was written, and because it was publicly available on the Web. We should point out that our intention was *not* to directly compare our interface to this particular interface. Instead, we are interested in comparing the general design between summarization using noun-adjective pairs (which are used primarily in RevMiner) and highlighting (which is used in our system).

Finally, we demonstrated the review-writing interface of our system, and asked the participants to write a review about a restaurant they recently visited using the interface. After they completed their reviews, we asked the participants whether they would use the interface to correct the feature-sentiment predictions when they were wrong.[9]

Participants

Thirteen college and graduate students (7 males and 6 females between the ages of 22 and 34) participated in this study. All of the participants read user reviews online at least once a month, and 8 of the 13 participants (62%) have experience in writing user reviews online. The study lasted approximately 30 minutes.

USER STUDY RESULTS

Highlighting helped participants digest user reviews

When comparing our interface with traditional review websites, 11 of the 13 participants (85%) suggested that the highlighting function helped them to understand more quickly the information contained in a massive amount of online reviews. One participant mentioned:

> *The highlighting function really helps me focus on the information I am interested in. I can get the information without spending time on the murmur of the reviewers.*

Although two participants thought this function didn't make significant differences, this result demonstrated that the feature-sentiment analysis and highlighting function were perceived as helpful for the majority of the participants in digesting the large number of user reviews.

[9]The exact wordings of our questions are listed in the appendix at the end of the paper.

More participants preferred highlighting over noun-adjective pair summarization

When participants were asked to compare our interface (highlighting) to RevMiner (noun-adjective pair), 6 of the 13 participants (46%) thought our interface was better, 3 of them (23%) thought RevMiner was better, and 4 of them (31%) thought it was a tie. The result suggests that about twice as many participants preferred the highlighting function, compared to those who preferred noun-adjective pair summarization. The reason was that the highlighting function allowed people to focus on the feature-sentiment information in its original context, which creates a good balance between focusing on some feature-sentiment information and the whole review. In contrast, the noun-adjective pair summarization allowed the participants to see only the compressed and fractured information, which was not easy for them to interpret. One of our participants noted:

> *I really like the first one (our interface) because it let me focus on some parts that I am interested in and I can also see its context. However, when reading reviews using the second one (RevMiner), I only see some short phrases. I have to first put them together to guess the meanings, so it takes more time and is hard to get the original context. This doesn't help me learn the experience of the previous users.*

The participants who favored noun-adjective pair summarization preferred it mainly because it offered more features than the three pre-defined features in our interface. However, this issue can be addressed by including more features in our system since the cost of adding new features is low.

Participants were motivated to correct erroneous predictions made by the system

After having the experience using our review-writing interface with real-time feature-sentiment analysis, 9 of the 13 participants (69%) expressed that they would correct errors made by the system. One of the participants mentioned,

> *When I write a review, I want to let others get my comments as clear as possible, so I care about the correctness of the information in the review and will correct the mistakes made by the system.*

Another participant said,

> *Because I read reviews using this system before, I know that my effort can help others understand my review, so I will provide the information even if it causes some extra work for me.*

This promising result demonstrates that the real-time feature-sentiment analysis does motivate users to correct the erroneous predictions of their own reviews. This is important as the system can collect more corrected labels to improve the classifier over time. We believe that the user's ability to see

immediately how their reviews will be classified is an important feature that motivates users to provide a low-cost (one click) easy correction of the automatic classification.

Of course, the current study cannot directly prove that most users really would correct the mistakes made by the system. Nevertheless, when the number of users increases, even if only a small portion of them provide feedback, the system still can benefit from the feedback and enhance classification accuracy.

Summary

To summarize, our user study answered the three questions related to our theses. First, the highlighting function provided by the proposed system can improve the user's reading experience. Second, highlighting is at least as good as, if not better than, noun-adjective pair summarization because it preserves the original context of the feature-sentiment information as expressed by the review writers. Finally, the interface with real-time feature-sentiment analysis can successfully motivate users to correct errors made by the system, so the classifiers behind the system can be improved as the number of users increases.

DISCUSSION

Reducing effort in order to motivate users to correct erroneous predictions

Our interface reduces the effort needed to provide feature-sentiment information by performing real-time analysis, which requires only that users correct some mistakes made by the system instead of segmenting, labeling, and rating their reviews themselves. As a result, our user study shows that around 70% of the participants expressed that they would provide corrected feature-sentiment information. However, about 30% of the participants did say that they would not correct the erroneous predictions made by the system. When asked, they said the effort required needed to be further reduced. As one of the participants explained:

> *I know that correcting the errors can be helpful to my readers, but I think it's just too much work for me. I need to click many icons there to make them right. That's why I choose to just ignore the errors.*

Therefore, it is important for us to design interfaces that allow users to correct the errors more easily. In future, we would like to experiment with different interface designs to determine how to motivate more users to provide the corrected data. One possible way is to design an interface that allows users to drag the icons and sentences directly. On the other hand, given that most users said they would provide the label and the classifier could benefit from the input, fewer and fewer corrections would be needed for future users as the classifier became more accurate.

Including more features in the proposed design

Our user study shows that about 23% of the participants preferred the intelligent interface that used noun-adjective pair summarization (RevMiner). As suggested by the participants, the biggest advantage of the interface is that it has more features that interest them. In contrast, our system has only three pre-defined features. This problem can be solved easily by including more features, so it is not an inherent limitation of the system. Since the efficiency of the classifiers is pretty high, adding more features will not cause any technical problems when more features are added. However, additional features may introduce a different problem. As mentioned earlier, maintaining a low level of effort required is important for motivating users to provide the correct labels. However, adding more features may increase the effort, as users need to remember what the possible categories are. This also may make the interface more complex. Therefore, there is clearly a tradeoff between providing more detailed categories and information and maximizing usability.

Real-time feature-sentiment analysis can encourage users to generate more structured reviews

Although our system was not intended to help users write reviews with higher quality, we did see that the real-time feature-sentiment analysis affected the reviews generated by the users. As one of the participants mentioned after she wrote a review using our interface:

> *The results of the (feature-sentiment) analysis let me know which part I haven't mentioned in my review, so I will try to write some words that are related to that part.*

In our user study, we also found that participants would try to write something related to the three pre-defined features before they finished. This shows that the real-time analysis can encourage users to generate reviews that cover more features, which can improve the quality of reviews collected by the system. A controlled experiment that shows how the real-time analysis affects review quality can be done in the future.

The upper bounds of classification accuracy

In our experiment, the food-classifier reached the highest F_1 score at 4000 training instances, and the service-classifier reached the highest at 3000 training instances; however, the F_1 score of the price-classifier continued to grow even after 5000 training instances. The intuitive explanation for this is that there were more positive training instances for food and service in our randomly chosen training dataset, so the classifiers learned faster initially.

In the future, it would be valuable to perform a study to determine the upper bounds of classification accuracy. Once a classifier reaches its upper bound, the system could stop asking users to provide feedback for that classifier. This can reduce the workload of the users of the system.

CONCLUSIONS AND FUTURE WORKS

In this paper, we presented a novel intelligent system that performs a two-layer feature-sentiment analysis in real time. The system can provide real-time predictions to users who are writing user reviews, which makes it very easy for them to provide feature-sentiment information by simply correcting the erroneous predictions. Our user study shows that about 70% of the participants were willing to correct the mistakes made by the system, which means that the proposed interface can successfully utilize the power of the crowd to collect a

massive amount of labeled data that can be used to train the supervised classifiers. Moreover, our experiment shows that the size of training data is positively correlated to the performance of the feature-sentiment analysis. As a result, we can expect that the analysis performed by the system can become more and more accurate as the number of system users increases.

In addition, we compared our system to existing intelligent interfaces with similar purposes. The results of our experiment show that the supervised method of our system can achieve much higher F_1 scores than traditional unsupervised methods can achieve. Moreover, our user study also shows that 46% of the participants preferred the highlighting function of our interface over the noun-adjective pair summarization, while only 23% of them preferred the summarization. This indicates that our system can provide more accurate feature-sentiment information and help users understand the information better than traditional interfaces with similar goals can.

The results of our experiment show that implicit crowdsourcing can be useful to improve supervised learning algorithm's ability to collect a huge amount of training data at no cost. The mechanism used in the proposed design also can be applied to other domains, like status updates in social media or contents in Q&A forums and is not limited to user reviews. However, there are still some limitations to the current design. One of the most essential issues is to find ways to further reduce the effort necessary for users to provide useful information. Moreover, it is also important to find a good way to include more features or even to let users input unspecified features themselves. We believe the work presented in this paper offers a good first step for more future studies that combine the strengths of intelligent interface and implicit crowdsourcing.

In the future, we would like to deploy our system in the wild to see if it really can help users and study how users interact with the system on a large scale. Furthermore, since the system involves its users to provide training data interactively, it is possible for us to include active learning [27] in our system design to further improve the performance of the supervised learning classifiers.

ACKNOWLEDGEMENTS

Many thanks to the anonymous reviewers for their valuable comments for us to improve the earlier draft of this work. We also would like to thank Siddharth Gupta for helping us implementing part of the system used in this study.

REFERENCES

1. Bernstein, M. S., Little, G., Miller, R. C., Hartmann, B., Ackerman, M. S., Karger, D. R., Crowell, D., and 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 '10, ACM (New York, NY, USA, 2010), 313–322.
2. Bigham, J. P., Jayant, C., Ji, H., Little, G., Miller, A., Miller, R. C., Miller, R., Tatarowicz, A., White, B., White, S., and Yeh, T. Vizwiz: nearly real-time answers to visual questions. In *Proceedings of the 23nd annual ACM symposium on User interface software and technology*, UIST '10, ACM (New York, NY, USA, 2010), 333–342.
3. Blei, D., and McAuliffe, J. Supervised topic models. In *Advances in Neural Information Processing Systems 20*, J. Platt, D. Koller, Y. Singer, and S. Roweis, Eds. MIT Press, Cambridge, MA, 2008, 121–128.
4. Callison-Burch, C., and Dredze, M. Creating speech and language data with amazon's mechanical turk. In *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*, CSLDAMT '10, Association for Computational Linguistics (Stroudsburg, PA, USA, 2010), 1–12.
5. Carenini, G., Ng, R. T., and Pauls, A. Interactive multimedia summaries of evaluative text. In *Proceedings of the 11th international conference on Intelligent user interfaces*, IUI '06, ACM (New York, NY, USA, 2006), 124–131.
6. Carenini, G., and Rizoli, L. A multimedia interface for facilitating comparisons of opinions. In *Proceedings of the 14th international conference on Intelligent user interfaces*, IUI '09, ACM (New York, NY, USA, 2009), 325–334.
7. Dave, K., Lawrence, S., and Pennock, D. M. Mining the peanut gallery: opinion extraction and semantic classification of product reviews. In *Proceedings of the 12th international conference on World Wide Web*, WWW '03, ACM (New York, NY, USA, 2003), 519–528.
8. Dong, R., McCarthy, K., O'Mahony, M., Schaal, M., and Smyth, B. Towards an intelligent reviewer's assistant: recommending topics to help users to write better product reviews. In *Proceedings of the 2012 ACM international conference on Intelligent User Interfaces*, IUI '12, ACM (New York, NY, USA, 2012), 159–168.
9. Esuli, A., and Sebastiani, F. Sentiwordnet: A publicly available lexical resource for opinion mining. In *In Proceedings of the 5th Conference on Language Resources and Evaluation (LREC06* (2006), 417–422.
10. Ho, C.-J., Chang, T.-H., Lee, J.-C., Hsu, J. Y.-j., and Chen, K.-T. Kisskissban: a competitive human computation game for image annotation. *SIGKDD Explor. Newsl. 12*, 1 (Nov. 2010), 21–24.
11. Howe, J. The rise of crowdsourcing. *Wired Magazine* (06 2006).
12. Hu, M., and Liu, B. Mining and summarizing customer reviews. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '04, ACM (New York, NY, USA, 2004), 168–177.

13. Huang, J., Etzioni, O., Zettlemoyer, L., Clark, K., and Lee, C. Revminer: an extractive interface for navigating reviews on a smartphone. In *Proceedings of the 25th annual ACM symposium on User interface software and technology*, UIST '12 (2012), 3–12.

14. Kearns, M. J., and Vazirani, U. V. *An introduction to computational learning theory*. MIT Press, Cambridge, MA, USA, 1994.

15. Kullback, S., and Leibler, R. On information and sufficiency. *The Annals of Mathematical Statistics 22*, 1 (1951), 79–86.

16. Lease, M., Carvalho, V., and Yilmaz, E., Eds. *Proceedings of the Workshop on Crowdsourcing for Search and Data Mining (CSDM) at the Fourth ACM International Conference on Web Search and Data Mining (WSDM)*.

17. Lease, M., Carvalho, V., and Yilmaz, E., Eds. *Proceedings of the ACM SIGIR 2010 Workshop on Crowdsourcing for Search Evaluation (CSE 2010)*. Geneva, Switzerland, July 2010.

18. Liu, B., Hu, M., and Cheng, J. Opinion observer: analyzing and comparing opinions on the web. In *Proceedings of the 14th international conference on World Wide Web*, WWW '05 (2005), 342–351.

19. Nichols, J., Mahmud, J., and Drews, C. Summarizing sporting events using twitter. In *Proceedings of the 2012 ACM international conference on Intelligent User Interfaces*, IUI '12, ACM (New York, NY, USA, 2012), 189–198.

20. Pang, B., and Lee, L. Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval 2*, 1–2 (2008), 1–135.

21. Pang, B., Lee, L., and Vaithyanathan, S. Thumbs up?: sentiment classification using machine learning techniques. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing - Volume 10*, EMNLP '02, Association for Computational Linguistics (Stroudsburg, PA, USA, 2002), 79–86.

22. Rashtchian, C., Young, P., Hodosh, M., and Hockenmaier, J. Collecting image annotations using amazon's mechanical turk. In *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*, CSLDAMT '10, Association for Computational Linguistics (Stroudsburg, PA, USA, 2010), 139–147.

23. Rijsbergen, C. J. V. *Information Retrieval*, 2nd ed. Butterworth-Heinemann, Newton, MA, USA, 1979.

24. Snow, R., O'Connor, B., Jurafsky, D., and Ng, A. Y. Cheap and fast—but is it good?: evaluating non-expert annotations for natural language tasks. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '08, Association for Computational Linguistics (Stroudsburg, PA, USA, 2008), 254–263.

25. Sorokin, A., and Forsyth, D. Utility data annotation with amazon mechanical turk. In *Computer Vision and Pattern Recognition Workshops, 2008. CVPRW '08. IEEE Computer Society Conference on* (june 2008), 1 –8.

26. Su, H., Deng, J., and Fei-Fei, L. Crowdsourcing annotations for visual object detection. In *Workshops at the Twenty-Sixth AAAI Conference on Artificial Intelligence* (2012).

27. Tong, S., and Koller, D. Support vector machine active learning with applications to text classification. *J. Mach. Learn. Res. 2* (Mar. 2002), 45–66.

28. Turney, P. D. Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, Association for Computational Linguistics (Stroudsburg, PA, USA, 2002), 417–424.

29. von Ahn, L., and Dabbish, L. Labeling images with a computer game. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, CHI '04, ACM (New York, NY, USA, 2004), 319–326.

30. von Ahn, L., and Dabbish, L. Designing games with a purpose. *Commun. ACM 51* (Aug. 2008), 58–67.

31. von Ahn, L., Liu, R., and Blum, M. Peekaboom: a game for locating objects in images. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '06, ACM (New York, NY, USA, 2006), 55–64.

32. Wang, H., Lu, Y., and Zhai, C. Latent aspect rating analysis on review text data: a rating regression approach. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '10, ACM (New York, NY, USA, 2010), 783–792.

33. 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 the SIGCHI Conference on Human Factors in Computing Systems*, CHI '11, ACM (New York, NY, USA, 2011), 1541–1550.

APPENDIX: QUESTIONNAIRE

Q1: When reading user reviews, do you prefer the first interface or the second interface? Why?

Q2: When reading user reviews, do you prefer the highlighting function of the second interface or the noun-adjective pair representation of the last interface? Why?

Q3: When writing user reviews on the review writing interface you just used, would you correct the erroneous predictions made by the system? Why or why not?