Weighing Stars: Aggregating Online Product Reviews for Intelligent E-commerce Applications

Zhu Zhang, *University of Arizona*

A new task in text-sentiment analysis adds usefulness scoring to polarity/opinion extraction to improve product-review ranking services, helping shoppers and vendors leverage information from multiple sources.

Human language is a medium not only for exchanging information but also for conveying subjective opinions and emotion. Recently, interest in text-subjectivity and sentiment analysis has increased as part of the larger research effort in affective computing, which aims to make computers understand and generate human-like emotions through language and other expressive activities such as gestures. A subset of this analysis relative to e-commerce involves mining the product reviews that thrive on the Web. Typical tasks include polarity prediction—distinguishing positive, negative, and neutral reviews—and opinion extraction. In essence, these tasks focus on "seeing stars" in text—a reference to the five-star scale system often used in rating systems—by predicting a text's semantic orientation. For example, a system based on these functions might predict the statement, "This is the most boring movie I've ever seen," to be a strongly negative comment.

However, researchers have largely overlooked another interesting dimension of this text-analysis problem. Online shoppers often wade through other people's reviews of a product to gauge their shopping decisions, and manufacturers also examine product reviews to monitor customer opinions and predict market trends. In both scenarios, the review readers seek relatively unbiased, informative evaluations of a given product by leveraging information from multiple sources, even though each individual review can be highly subjective. More specifically, the review readers must not only make sense of each individual opinion but also weigh multiple—sometimes conflicting—views.

To readers who are thinking along this line, product reviews aren't equally useful, regardless of the polarity of embedded opinions. In this article, I will present a study that identifies a new task in ongoing text-sentiment analysis research to aggregate online product reviews. The new task focuses on "weighing stars" in light of two orthogonal dimensions: polarity/opinion extraction and usefulness scoring. I view usefulness scoring as a regression-analysis problem, and models built on a diverse feature set computed from review text achieved promising performance on four Amazon product review collections.

Motivation for weighing stars

Let's consider a customer review from Amazon.com for a digital camera, the Canon Powershot SD300. While the following review encompasses mixed feelings about the product, 170 out of 178 Amazon shoppers found it helpful. In other words, they considered it to be informative:

> Handling: The interface is similar to that in other Canon digital compacts, which helps your learning curve. The case is in metal, except the USB and the battery cover. They are both made of plastic and feel very fragile. The metal tripod mount is located closely below the lens. The LCD screen is reasonably viewable under daylight conditions.

1541-1672/08/$25.00 © 2008 IEEE
Published by the IEEE Computer Society

> Canon celebrated that SD300 is the first compact that uses the DIGIC II processing, and with my experience so far, the camera does respond faster when compared with DIGIC based ones. It however appears to be on par with recent Sony and Olympics models. I did not measure the various response times scientifically however.
>
> Picture quality: Contrary to other comments, I was not “blown away” by quality of the pictures. The lens produces not serious, but significant purple edges in bright sunlight, and shows problems with dark corners like other compacts. Color production is rich with high contrast, a big plus. Sadly I find the pictures appear noisy when taking under low light conditions. I suspect either I have a faulty unit, or there are some design issues?
>
> Complaints: Can’t review picture histogram easily (two to three steps). Noisy operations. No battery level indicator.
>
> So far I find the SD300 a good and decent pocket camera. However in the same market (similar specs and price) there are many other choices, and the SD300 does not excel specifically in any area.

By contrast, we can easily imagine product reviews as simple as

> X is the greatest product I’ve ever seen.

or

> Product Y sucks. I hate every single aspect of it.

These reviews certainly encode strong polarity and reflect their authors’ strong opinions. However, they’re not particularly reliable or useful in informing readers’ shopping decisions.

> Product reviews can encode strong polarity and reflect strong author opinions without being particularly reliable or useful in informing readers‘ shopping decisions.

When presented with a mixed bag of positive and negative, useful and not-so-useful reviews, how does an online shopper leverage such diverse evidence? Or in technical terms, how should we design a “review aggre-bot” that crawls several online sources and generates overall product ratings through aggregation of multiple reviews? Assuming explicit numerical ratings are available, some e-commerce sites currently perform trivial aggregation by simply averaging numerical ratings from multiple customers. The aggregation is further limited to the ratings at the single site, not across multiple sites.

As a motivating thought at a very general level, we can envision the following weighted-average framework that enables nontrivial aggregation of nonnumerical product evaluations:

$$V(P)=\frac{\sum_{i=1}^{n}\left(u\left(T_i(P)\right)*Polarity\left(T_i(P)\right)\right)}{\sum_{i=1}^{n}u\left(T_i(P)\right)} \qquad (1)$$

in which the overall valuation V of a product P is a weighted average of the polarity of each individual review $T_i(P)$, where the weights indicate their relative usefulness. We can view this integration framework as a simplified version of the generalized-opinion-pooling problem studied in statistics,[1] where the pooling operator is a simple linear combination. Whenever applicable, the usefulness score $u(T_i(P))$ can be employed as a ranking function to help eliminate information overload for review readers.

Text subjectivity and polarity

Product reviews are inherently subjective, and their implications for business activities make them a popular target for polarity analysis and opinion extraction research.

Subjectivity in text

In natural language, subjectivity refers generally to language aspects that express opinions, evaluations, and speculations. Statistical algorithms can learn and generalize subjective language from text corpora. Janyce Wiebe and her colleagues provide a good overview of research in this line.[2] In their own research, they generated and tested subjectivity clues, including low-frequency words, collocations, and adjectives and verbs identified using distributional similarity. They examined the features working together in concert. In addition, they showed that the density of subjectivity clues in the surrounding context strongly affects how likely it is that a word is subjective. Finally, they used the subjectivity clues to identify opinionated chunks in text.

Hong Yu and Vasileios Hatzivassiloglou approached the problem of separating opinions from fact, at both the document and sentence levels.[3] They presented a naive Bayes classifier for discriminating documents with a preponderance of opinions, such as editorials, from regular (fact-based) news stories, and they described three unsupervised statistical techniques to detect opinions at the sentence level. They also developed a model for sentence-level polarity classification (positive, negative, and neutral) that combined evidence from multiple types of semantically oriented words.

Instead of viewing subjectivity as a sentence- or passage-level property and studying a text’s overall polarity, Theresa Wilson, Janyce Wiebe, and Paul Hoffmann focused on phrase-level sentiment analysis.[4] For example, there are two polar phrases in the sentence “We *don’t hate*$^+$ the sinner, but we *do hate*$^-$ the sin.” In a two-stage process, the authors first determine an expression or phrase to be neutral or polar; then, if the phrase is polar, the process further disambiguates it into different contextual polarity classes (positive, negative, both, and neutral). With this approach, a system could automatically identify the contextual polarity for a large subset of sentiment expressions, achieving competitive results.

Researchers have recognized the importance of acquiring lexical clues for subjectivity analysis, so they’ve created a number of reusable resources, which I employ in this study.

Mining polarity and opinions in product reviews

In an early paper in this domain, Peter Turney presented a simple unsupervised learning algorithm for classifying reviews as “recommended” (thumbs up) or “not recommended” (thumbs down).[5] Specifically, the algorithm first calculated a phrase’s semantic orientation to be the mutual information between the given phrase and the word “excellent,” minus the mutual information between the given phrase and the word “poor.” In turn, it predicted a review’s overall polarity by the average semantic orientation of review text phrases that contained adjectives or adverbs.

Using Internet Movie Database review data (http://reviews.imdb.com/Reviews), Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan also considered the problem of classifying documents by overall sentiment—for example, whether a review is positive or negative.[6] They showed that standard machine learning techniques outperform human-produced baselines. However, the three machine learning methods employed (naive Bayes, maximum-entropy classification, and support vector machines) don’t perform as well on sentiment classification as on traditional topic-based categorization because of challenges related to noncompositional semantics and discourse structures.

Pushing further along the same line, Bo Pang and Lillian Lee attempted to approach the rating-inference problem.[7] In this problem, rather than simply deciding whether a review is “thumbs up” or “thumbs down,” you must determine an author’s evaluation with respect to a multipoint scale (for example, one to five stars). In this work, Pang and Lee compared a metric-labeling approach with both multiclass and regression versions of support vector machines (SVMs).

In anticipation of a review-aggregation service, I aim to build a computational model that predicts a review’s usefulness by exploiting its linguistic properties.

A product usually has multiple features that different customers might evaluate differently. Shifting from classification to extraction, Ana-Maria Popescu and Oren Etzioni introduced Opine, an unsupervised information-extraction system.[8] Opine mines reviews to build a model of important product features, their evaluation by reviewers, and their relative quality across products.

In a similar effort, Bing Liu, Minqing Hu, and Junsheng Cheng proposed a framework for analyzing and comparing consumer opinions of competing products.[9] They also implemented a prototype system called Opinion Observer. The system offered visualization functionality that let users clearly see each product’s feature strengths and weaknesses according to consumer opinions. The system used supervised rule-discovery techniques to extract product features and corresponding pros and cons.

With a somewhat different focus, Soo-Min Kim and Eduard Hovy presented a system that automatically extracted pro and con reasons from the review text.[10] The reasons might themselves be in the form of either fact or opinion. The authors trained a maximum-entropy model to solve the sentence-classification problem.

A comparison of one object with another offers another interesting perspective on product review or evaluation—for example, “Feature f of product X is not as good as that of product Y.” Comparisons can be subjective or objective. Practically speaking, direct comparisons can be one of the most convincing evaluations and might therefore be even more important than opinions on individual objects.

Almost all prior research in product-review mining has focused on the “seeing stars” dimension—that is, polarity prediction and opinion extraction. Only recently have researchers started examining the usefulness dimension. Parallel to and independent of my work, Soo-Min Kim and her colleagues investigated the problem of assessing review helpfulness and showed that review length, unigrams, and product ratings are strong predictors.[11] Besides showing different empirical results on usefulness prediction, my study makes unique contributions by bringing in an aggregation perspective beyond the ranking perspective.

Predicting usefulness with statistical regression models

In anticipation of a review-aggregation service as outlined by Equation 1, I aim to build a computational model that predicts a review’s usefulness by exploiting its linguistic properties. I distinguish usefulness—in the sense of quality, reliability, or informativeness—from indifference. Totally indifferent or neutral reviews are useless, whereas well-grounded subjective opinions can be convincing and illuminating. In other words, a product review’s usefulness is orthogonal to its polarity or embedded opinions.

Strictly speaking, as in economics, a product’s utility is consumer dependent. In this study, however, as a first approximation, I base the notion of a review’s usefulness on the perception of a statistically average reader or shopper. This is plausible because the ultimate goal is to reduce the information overload for online shoppers by ranking or aggregating reviews.

I assign a real-valued usefulness score to each review, which makes the scoring problem a natural regression problem. Formally, given a product review T, I compute a number of features $f_1(T), \ldots, f_j(T), \ldots, f_P(T)$. The task is to approximate a function

$$u(T) = F(f_1, \ldots, f_j, \ldots, f_P)$$

such that the output $u \in [0, 1]$ and reflects the real usefulness of T as accurately as possible.

We can now discuss two aspects of the statistical learning framework: the learning (regression) algorithms and the features (independent variables).

Learning algorithms

I selected regression algorithms from the SVM family, which represents the state of the art in statistical learning. Although researchers have frequently applied them to classification problems, they can also be formulated to take on regression tasks.

SVM-based algorithms—in this case, support vector regression (SVR) algorithms—are attractive because they employ powerful kernel functions that can capture structural patterns in data. In the current study, I use the popular radial-basis-kernel function (RBF), which handles the potentially nonlinear relationship between the target value and features.

I experiment with two types of SVR algorithms, which are based on different formulations of the optimization problem:

- ϵ-support vector regression (ϵ-SVR)
- ν-support vector regression (ν-SVR)

In both cases, I apply the original algorithms as they’re implemented in the machine learning package Libsvm (www.csie.ntu.edu.tw/~cjlin/libsvm).

Usefulness features

Generally speaking, a good product review should be a "reasonable" mixture of subjective valuation and objective information, such that the text is trustworthy and informative. The feature space in the statistical learning framework ought to capture this intuition.

In this study, given a product review text *T*, I compute three types of features (or "clues") that can potentially help distinguish useful and useless reviews: lexical similarity features, shallow syntactic features, and lexical subjectivity clues.

Lexical similarity features (LexSim). Clearly, a useful or informative customer review shouldn't be a literal copy or a loyal rephrase of the product specification *S*. It should reflect the customer's experience with or evaluation of the product, not the manufacturer's description or expectation. On the other hand, a good review is supposed to base subjective judgment on objective observation. It should therefore echo the product specification to a reasonable extent, although possibly in a positive or negative tone.

A similar situation is conceivable between a customer review and an editorial review *E*, the latter of which approximates a relatively objective and authoritative view of the product.

With these motivations in mind, I measure the similarity between customer review and product specification, *sim*(*T*, *S*), and that between customer review and editorial review, *sim*(*T*, *E*).

I use the standard cosine similarity in vector space model, with tf-idf (term frequency-inverse document frequency) weighting, as defined in information-retrieval literature.

Three types of computable features can help distinguish useful from useless review texts: lexical similarites, shallow syntactic features, and lexical subjectivity clues.

Shallow syntactic features (ShallowSyn). Is it the case that relatively useful reviews tend to exhibit certain syntactic properties? We can characterize a text *T*'s linguistic styles at a shallow syntactic level (relative to other "deep" syntactic structures such as parse trees) by computing the number of words with each of the following part-of-speech tags:

- *Proper nouns*: references to existing, maybe technical, concepts.
- *Numbers*: tendencies of quantification.
- *Modal verbs*: reflections of certainty, confidence, mood, and other such instances of modality.
- *Interjections*: signals of emotion.
- *Comparative and superlative adjectives*: indicators of comparison.
- *Comparative and superlative adverbs*: indicators of comparison in superlative forms.
- *Wh-determiners, wh-pronouns, possessive wh-pronouns,* and *wh-adverbs*: signifiers of either questions or other potentially interesting linguistic constructs such as relative clauses.

This study also computes simple counts such as the number of words and number of sentences in the review text *T*, and uses them as features.

Lexical subjectivity clues (LexSubj). An interesting set of features takes advantage of lexical resources created in prior text-sentiment analysis research to capture the subjectivity-objectivity mixture at a lexical semantic level. Specifically, lexical subjectivity clues involve counting the review text words that belong to the following lists:

1. The list of subjective adjectives learned in Wiebe.[12]
2. The list of subjective adjectives learned in Hatzivassiloglou and Wiebe.[13] More specifically, this list contains
   - dynamic adjectives (for example, "careful" and "serious").
   - polarity *plus* adjectives (for example, "amusing"), manually or automatically identified. (The manually identified words constitute one feature, and the automatically another feature. The distinction is similar for items below.)
   - polarity *minus* adjectives (for example, "awful"), manually or automatically identified.
   - gradability *plus* adjectives (for example, "appropriate"), manually or automatically identified.
   - gradability *minus* adjectives (for example, "potential"), manually or automatically identified.
3. The lists of strong subjective nouns (for example, "domination" and "evil") and weak subjective nouns (for example, "reaction" and "security") generated by the MetaBoot algorithm.[14]
4. The lists of strong and weak subjective nouns generated by the Basilisk algorithm.[15]

These word lists essentially specify a reduced feature space for constructing a word-based feature vector.

For the learned model to generalize well, I won't count each individual word's frequency. Instead, I count only the total occurrences of words in each list. For example, if the review text *T* contains five weak subjective nouns, I give a value "5" to the feature "WeakSubjNoun" instead of assigning the value "1" to five binary features.

In total, I have 14 word lists for the study and therefore 14 features in this category.

Experimental data collection and treatment

Although a number of online-shopping sites such as CNET offer potential sources for customer review data, Amazon has a relatively convenient set of APIs, Amazon Web Services (http://aws.amazon.com/ecs), through which you can access various product data including customer reviews.

I downloaded reviews and related data in three different domains: electronics, video, and books. For electronics, I used keyword-based search to identify Canon products (I named the corresponding collection *Canon*) and Sony products (*Sony*). For books, I used keyword-based search again to identify engineering books (*Engineering*), and I used the "AudienceRating" field to retrieve PG-13 movies (*PG-13*).

Table 1. Amazon customer-review statistics.

Collection	No. of reviews	No. of authors
Canon	2,394	2,384
Sony	3,916	3,891
Engineering	6,120	6,032
PG-13	11,543	11,445

Table 2. Statistics of data sets after treatment.

Collection	Mean	Standard deviation	No. of reviews
Canon	0.7914	0.2839	624
Sony	0.7750	0.2821	775
Engineering	0.7531	0.2992	1,255
PG-13	0.5605	0.3756	654

Table 3. Correlation between usefulness score and review length.

Collection	r^2
Canon	0.0042
Sony	0.0585
Engineering	0.0997
PG-13	0.0853

Table 4. Correlation between usefulness score and rating.

Collection	r^2
Canon	0.0352
Sony	0.1072
Engineering	0.0615
PG-13	0.1057

I developed a set of Perl scripts to automate data collection and then collected the following data:

1. Customer reviews for the products whose Amazon Standard Identification Number (ASIN) appears in the search response to queries for the four selected product categories.
2. Product metadata, including both product descriptions and Amazon editorial reviews, corresponding to the four product categories.
3. All reviews written by customers who also reviewed the four selected product categories. This is to facilitate later author-based analysis.

Table 1 summarizes basic statistics for the four collections. Amazon Web Services (AWS) restricts the number of search results to a maximum of 2,500 records per query. This limit constrains the data collection.

With regard to data treatment, the only component I haven't yet specified is how to acquire the regression model's target value, given a review text *T*. In other words, we still need to operationalize the gold-standard definition of $u(T)$.

Almost every Amazon customer review comes with a vote regarding its usefulness ("*x* out of *y* people found the following review helpful"). This actually provides a direct and convenient way to approximate a gold-standard usefulness value for a given review. Formally, I define usefulness, *u*, as

$$u = \frac{x}{y}$$

In fact, in real online systems, this is how you directly obtain the usefulness score. But a significant time lag exists between someone writing a review and people rating it. A system, such as the one I propose, tries to get around this lag by approximating the usefulness score in an almost real-time fashion.

In my experiments, I consider only the reviews with more than 10 votes ($y > 10$) to be valid. This ensures the regression model's robustness.

Table 2 summarizes the distribution of review usefulness scores in the four Amazon collections as well as their total number of distinct reviews after filtering out duplicates and those with fewer than 10 votes.

Usefulness scoring of product reviews

Given a learned scoring function, we can evaluate its quality using two standard metrics for regression analysis—namely, the squared correlation coefficient

$$r^2 = \frac{\left(\sum_{i=1}^{n}\left(u_i - \bar{u}\right)\left(\hat{u}_i - \bar{\hat{u}}\right)\right)^2}{\sum_{i=1}^{n}\left(u_i - \bar{u}\right)^2 \sum_{i=1}^{n}\left(\hat{u}_i - \bar{\hat{u}}\right)^2}$$

and the mean-squared error

$$\sigma^2 = \frac{1}{n}\sum_{i=1}^{n}\left(u_i - \hat{u}_i\right)^2$$

where u_i and $\hat{u}_i$ are the real and predicted usefulness scores, respectively, and $\bar{u}$ and $\bar{\hat{u}}$ represent the mean of the corresponding sample.

Two baseline models

Before I present the regression results, let's consider the relationship between a review's usefulness and its length. You might intuitively expect a strong positive correlation between these phenomena—in other words, the longer a review is, the more useful it is.

However, Table 3 shows that the correlation, although positive, is quite weak. (All linear correlation coefficients in Table 3 are positive; I present the squared versions for comparison with regression results later on. Review length is measured by number of words.) Because the correlation is weak, predicting usefulness scores requires building nontrivial regression models.

You might also intuitively expect a review's usefulness to cor-

Table 5. Summary of usefulness regression results.*

Product	Feature Set	ϵ-Support Vector Regression		ν-Support Vector Regression	
		r^2	σ^2	r^2	σ^2
Canon	LexSim	0.0049	0.0957	0.0044	0.0853
	ShallowSync	0.2726	0.0601	0.2796	0.0591
	LexSubj	0.0448	0.0902	0.0315	0.0819
	All	0.3028	0.0565	**0.3214**	**0.0547**
Sony	LexSim	0.0077	0.0918	0.0046	0.0828
	ShallowSync	0.1896	0.0673	0.1874	0.0663
	LexSubj	0.0561	0.0876	0.0481	0.0784
	All	0.2451	0.0607	**0.2466**	**0.0610**
Engineering	LexSim	0.0216	0.0947	0.0103	0.0915
	ShallowSync	0.3128	0.0615	0.3205	0.0623
	LexSubj	0.0674	0.0907	0.0589	0.0876
	All	0.3514	0.0581	**0.3577**	**0.0585**
PG-13 movies	LexSim	0.0014	0.1484	0.0036	0.1406
	ShallowSync	0.4176	0.0829	**0.4219**	**0.0831**
	LexSubj	0.0412	0.1479	0.0441	0.1351
	All	0.4145	0.0826	0.4147	0.0826

*Bold font indicates the most competitive results in each product category.

relate strongly correlation with its polarity—that is, with the number of stars associated with the review. However, Table 4 shows that the correlation is only moderate. (Again, all linear correlation coefficients are positive; I present the squared versions for comparison with regression results later on.) These results are comparable to those of Kim and her colleagues.[11] However, a univariate polarity-based model is far less powerful than the full-regression model I will build later in this study.

Does authorship matter?

Intuitively, you might expect a product review's quality to be highly dependent on who wrote it. We expect authors' knowledge (particularly for the products being reviewed) as well as personal background, habits, preferences, communication skills, and so on, to play important roles when they write reviews. To verify this hypothesis, I was able to identify six prolific authors in the data. (The AWS downloading constraint limited my ability to find such authors. On the other hand, to perform the analysis I report in this subsection, a larger number of authors would have required statistical techniques beyond the analysis of variance (ANOVA). Each of the six authors has written at least 10 qualifying reviews (reviews having more than 10 votes) across multiple product categories; in total, the six authors have written 221 reviews.

Using this data set, a one-way ANOVA analysis between the usefulness score and the author suggests a very strong influence of the latter ($F = 15.729$, $p << 0.001$). So authorship does seem to be a powerful usefulness predictor. Arguably, it's the real predictor.

Unfortunately, without explicit author profiling, measuring authors' intrinsic properties is almost impossible. Even if we could measure the properties is almost impossible, even Shakespeare doesn't always produce masterpieces. Therefore, I base my prediction model on features extractable from the review text.

Regression results and discussion

Table 5 summarizes the regression performance on the four review collections (*Canon*, *Sony*, *Engineering*, and *PG-13*). All the results in this section are based on 10-fold cross-validation**,** with parameter optimization using grid search.

For all four product categories, the rows represent different feature sets ("ALL" stands for the full-feature vector); the columns correspond to the performance of the ϵ-SVR and ν-SVR algorithms, measured by squared correlation coefficient r^2 and mean squared error σ^2 respectively. Bold font indicates the most competitive results in each product category**.**

On the basis of these experimental results, I have the following observations:

- Across all four collections, the regression results are rather similar qualitatively, although it does seem that the SVR models do a better prediction job on engineering books than on Sony products. In all cases, the regression models significantly outperform the length-based and rating-based baselines.
- On average, the strongest regression model for each collection achieves r^2 of 0.3351 and $\sigma^2 < 0.10$, which implies a reasonably strong linear correlation (about 0.5789) between the predicted and gold-standard usefulness scores. On the other hand, given the cognitive complexity involved in the perception of usefulness, it's not surprising that regression models based on machine-

Table 6. Rank correlation between aggregated review polarity and Amazon sales rank.

Data set	Spearman's ρ			Kendall's τ		
	SimAvg	PredAvg	GsAvg	SimAvg	PredAvg	GsAvg
Canon	0.1023	0.1532	0.1412	0.0768	0.1052	0.0962
Sony	0.0155	0.0313	0.0443	0.0104	0.0297	0.0359
Engineering	0.0780	0.0910	0.0967	0.0569	0.0654	0.0674
PG-13	0.0971	0.1312	0.1504	0.0753	0.0899	0.1013

Note: SimAvg = simple average; PredAvg = weighted average based on predicted usefulness score; GSAvg = weighted average based on gold-standard usefulness.

computable features capture only part of the picture.

- Between the two different regression algorithms, ν-SVR outperforms ϵ-SVR in all four collections, although not significantly.
- Looking at the feature vector for the regression models, the three feature groups have different effects on the final output.

With regard to the last observation, the LexSim feature set plays a very minor role in the regression model. Intuitively, how useful a review is doesn't necessarily correlate with how similar it is to the corresponding product specification or authoritative review. Instead, a review's usefulness is based on properties inherent in the review itself. The LexSubj clues have limited influence on the usefulness scoring. This means a product review's perceived usefulness barely correlates with the subjectivity or polarity embedded in the text. It also justifies the orthogonal view presented at the beginning of this article. Finally, the ShallowSyn feature set accounts for most of the regression model's predictive power. This phenomenon demonstrates that highly useful reviews do stand out because of the linguistic styles in which they're written. It's also consistent with the finding on the significance of authorship, because writing styles usually reflect properties inherent in the writers' personalities.

These insights differ from those of Kim and her colleagues.[11] Although the nonidentical data sets, data-treatment techniques, and feature sets could possibly account for the differences, I believe my findings are in better alignment with people's actual online shopping experience. For example, my first baseline model showed that review length isn't a powerful predictor, and we know intuitively that a useful review doesn't have to be lengthy. Short reviews could have substantially helpful information.

The second baseline model confirmed that the product rating has a reasonable correlation with review usefulness. However, numerical ratings aren't always available in real online systems. The lexical subjectivity clues, which are intended as proxies for review polarity, turn out to be not particularly powerful in the regression models. On one hand, this echoes the fact that "seeing stars" in text is still a challenging research problem; on the other hand, it suggests the necessity of seeking other easy-to-compute predictors. Unlike Kim and her colleagues,[11] I showed that easily computable syntactic features are the strongest predictors for review usefulness. I believe this is more meaningful in real life because a review's quality is grounded in the author's expertise and other intrinsic properties, reflected in the writing itself. In contrast, opinion polarity is typically not a reliable indicator of how useful a review is because positive reviews can be fishy and negative reviews can be informative.

I hypothesized authorship as the real predictor variable behind review usefulness and provided preliminary empirical support. This lends further credence to the contrast between syntactic and polarity features. While the writing style and knowledge level of an author (syntactic features being their proxies) are relatively stable, opinions and sentiments (lexical subjectivity clues being their proxies) are usually situational. The syntactic predictors thus naturally dominate the resultant regression models.

Review aggregation

As mentioned earlier, we can employ a review's usefulness score as a ranking function but, more important, as a weight in an aggregation mechanism, such as Equation 1. The hypothesis is that, given a product, a nontrivial aggregation of customer opinions (by considering the quality or usefulness of reviews) is a better predictor for overall market response to a product than some trivial form of aggregation (such as the simple arithmetic averaging typically implemented in many online shopping sites).

To validate the hypothesis, I measured the correlation between aggregated review polarity and sales performance. More specifically, given a product P and its customer reviews T_i, $i = 1 \dots n$, I used P's sales rank $Rank(P)$ as the proxy for its sales performance, which is by far the closest approximation available in Amazon data. I used the number of "stars" associated with each review to instantiate its inherent polarity—that is, $Polarity(T_i(P))$—and I measured the rank correlation—specifically, the well-known Spearman's ρ and Kendall's τ—between sales rank and aggregated review polarity.

We can consider and compare three types of review aggregation, all following the framework in Equation 1:

- *SimAvg* computes the aggregated polarity as a simple average of individual review polarity; that is, $u(T_i(P))$ is always 1.
- *PredAvg* computes the aggregated polarity as a weighted average of individual review polarity; the weight assigned to each review T_i is its predicted usefulness score $\hat{u}(T_i(P))$.
- *GsAvg* computes the aggregated polarity again as a weighted average of individual review polarity; the weight assigned to each review T_i is its gold-standard usefulness score $u(T_i(P))$.

Table 6 summarizes the results on the four data sets. In the experiments, I computed the predicted usefulness score $\hat{u}(T_i(P))$ by applying an ν-SVR model trained on all valid reviews other than T_i. We see that in all cases, regardless of the correlation measure, the aggregated polarity using the gold-standard usefulness scores—that is, *GsAvg*—is always more strongly correlated with sales rank than the simple aggregation (*SimAvg*) is. The performance of predicted-usefulness-based aggregation (*PredAvg*) falls between the two, with the exception of the Canon data, on which *PredAvg* even slightly outperforms *GsAvg*. Therefore, my hypothesis is well supported, which necessitates the building of intelligent review-aggregations services in e-commerce sites.

On the other hand, the correlation values in Table 6 aren't very high. One possible reason is that an online shopper's purchase decisions involve many factors, just as any complicated decision process does. So, the aggregated evaluation derived from product reviews, whatever the aggregation mechanism is, is only one variable among many that together constitute a model explaining or predicting a market response to a certain product.

Another reason might stem from the four data sets all having quite diverse product items. Thus, the sales-rank data reflects performance only at a relatively coarse level. On tighter data collections, where products are more similar, sales should more strongly correlate with user opinions. In such collections, factors such as product functionality and pricing range are relatively better controlled.

Several challenging research questions remain open in text-sentiment mining. For example, the author-based analysis presented in this study is still very preliminary. A review's usefulness depends on not only its author's intrinsic character but also the author's expertise in the product category. Factorial experiments based on usefulness as a function of author and product category were infeasible because Amazon's query limit kept the data sparse. On the other hand, explicit author (shopper) profiling to incorporate information such as demographic data and shopping history should improve the regression model's predictive power by giving it more useful features.

In addition, I conducted this study's aggregation experiments on reviews from a single site, Amazon.com. An immediate future extension is to build similar models and perform experiments on product reviews from multiple Web sites. In the long run, an integrated framework more general than Equation 1 is certainly desirable. The framework should consider multiple aspects of text-sentiment mining in concert, including polarity classification, opinion extraction, usefulness scoring, comparison mining, and so on.

Other text genres on the Web, such as blogs and forums, are also rich in sentiments and opinions about events, public figures, and social movements. Mining and aggregating sentiments from these sources offers a new playground for the Web and text-mining research community. This study is a first attempt to present the aggregation point of view.

The Author

Zhu Zhang is an assistant professor in the Department of Management Information Systems at the University of Arizona. His research interests include data and text mining, machine learning, and Internet computing. Zhang received his PhD in computer and information science from the University of Michigan. He's a member of the ACM, AAAI, and Association for Computational Linguistics. Contact him at zhuzhang@u.arizona.edu.

References

1. A. Garg, T. S. Jayram, S. Vaithyanathan, and H. Zhu, "Generalized Opinion Pooling," *Proc. 8th Int'l Symp. AI and Math.*, *Ann. Math. and AI*, http://rutcor.rutgers.edu/~amai/aimath04.
2. J. Wiebe et al., "Learning Subjective Language," *Computational Linguistics*, vol. 30, no. 3, 2004, pp. 277–308.
3. H. Yu and V. Hatzivassiloglou, "Towards Answering Opinion Questions: Separating Facts from Opinions and Identifying the Polarity of Opinion Sentences," *Proc. 2003 Conf. Empirical Methods in Natural Language Processing* (ACL 03), Assoc. for Computational Linguistics, 2003, pp. 129–136.
4. T. Wilson, J. Wiebe, and P. Hoffmann, "Recognizing Contextual Polarity in Phrase-Level Sentiment Analysis," *Conf. Empirical Methods in Natural Language Processing* (ACL 05), Assoc. for Computational Linguistics, 2005, pp. 347–354.
5. P.D. Turney, "Thumbs Up or Thumbs Down? Semantic Orientation Applied to Unsupervised Classification of Reviews," *Proc. 40th Ann. Meeting Assoc. for Computational Linguistics* (ACL 02), Assoc. for Computational Linguistics, 2001, pp. 417–424.
6. B. Pang, L. Lee, and S. Vaithyanathan, "Thumbs Up? Sentiment Classification Using Machine Learning Techniques," *Proc. Conf. Empirical Methods in Natural Language Processing* (EMNLP 02), Assoc. for Computational Linguistics, 2002, pp. 79–86.
7. B. Pang and L. Lee, "Seeing Stars: Exploiting Class Relationships for Sentiment Categorization with Respect to Rating Scales," *Proc. 43rd Ann. Meeting Assoc. for Computational Linguistics* (ACL 05), Assoc. for Computational Linguistics, 2005, pp. 115–124.
8. A.M. Popescu and O. Etzioni, "Extracting Product Features and Opinions from Reviews," *Conf. Empirical Methods in Natural Language Processing* (EMNLP 05), Assoc. for Computational Linguistics, 2005, pp. 339–346.
9. B. Liu, M. Hu, and J. Cheng, "Opinion Observer: Analyzing and Comparing Opinions on the Web," *Proc. 14th Int'l. Conf. World Wide Web* (WWW 05), ACM Press, 2005, pp. 342–351.
10. S.-M. Kim and E. Hovy, "Automatic Identification of Pro and Con Reasons in Online Reviews," *Proc. COLING/ACL on Main Conf. Poster Sessions*, Assoc. for Computational Linguistics, 2006, pp. 483–490.
11. S.-M. Kim et al., "Automatically Assessing Review Helpfulness," *Proc. 2006 Conf. Empirical Methods in Natural Language Processing* (EMNLP 06), Assoc. for Computational Linguistics, 2006, pp. 423–430.
12. J. Wiebe, "Learning Subjective Adjectives from Corpora," *Proc. 15th Nat'l Conf. Artificial Intelligence* (AAAI 00), AAAI Press, 2000, pp. 735–740.
13. V. Hatzivassiloglou and J.M. Wiebe, "Effects of Adjective Orientation and Gradability on Sentence Subjectivity," *Proc. 18th Conf. Computational Linguistics*, Assoc. for Computational Linguistics, 2000, pp. 299–305.
14. E. Rilo, J. Wiebe, and T. Wilson, "Learning Subjective Nouns Using Extraction Pattern Bootstrapping," *Proc. Conf. Computational Natural Language Learning* (CoNLL 03), 2003, pp. 25–32.
15. M. Thelen and E. Rilo, "A Bootstrapping Method for Learning Semantic Lexicons Using Extraction Pattern Contexts," *Proc. Conf. on Empirical Methods in Natural Language Processing* (EMNLP 02), Assoc. for Computational Linguistics, 2002, pp. 214–221.