REVRANK: a Fully Unsupervised Algorithm for Selecting the Most Helpful Book Reviews

Oren Tsur
Institute of Computer Science
The Hebrew University
Jerusalem, Israel
`oren@cs.huji.ac.il`

Ari Rappoport
Institute of Computer Science
The Hebrew University
Jerusalem, Israel
`www.cs.huji.ac.il/~arir`

Abstract

We present an algorithm for automatically ranking user-generated book reviews according to review helpfulness. Given a collection of reviews, our REVRANK algorithm identifies a lexicon of dominant terms that constitutes the core of a virtual optimal review. This lexicon defines a feature vector representation. Reviews are then converted to this representation and ranked according to their distance from a 'virtual core' review vector. The algorithm is fully unsupervised and thus avoids costly and error-prone manual training annotations. Our experiments show that REVRANK clearly outperforms a baseline imitating the Amazon user vote review ranking system.

Introduction

The World Wide Web contains a wealth of opinions on just about anything. Online opinions come in various forms and sizes, from short and informal talkbacks, through opinionated blog postings to long and argumentative editorials. An important source of information are postings in internet forums dedicated to product reviews. In this era of user generated content, writing product reviews is a widespread activity. People's buying decisions are significantly influenced by such product reviews. However, in many cases the number of reviews is rather large (there are many thousands of reviews on the popular products), which causes many reviews to be left unnoticed. As a result, there is an increasing interest in review analysis and review filtering, with the goal of automatically finding the most helpful reviews.

In order to help users find the best reviews, some websites (e.g., amazon.com) employ a voting system in which users can vote for review helpfulness ("was this review helpful to you? yes/no"). However, user voting mechanisms suffer from various types of bias (Liu et al. 2007), including the imbalance vote bias (users tend to value others' opinions positively rather than negatively), the winner circle bias (reviews with many votes get more attention therefore accumulate votes disproportionately), and the early bird bias (the first reviews to be published tend to get more votes).

In this paper we describe a novel method for content analysis, which is especially suitable for product reviews. Our system automatically ranks reviews according to their estimated helpfulness. First, our REVRANK algorithm identifies a core of dominant terms that defines a virtual optimal review. This is done in two stages: score terms by their frequency, and then identify the terms that are less frequent but contribute more information that is relevant to the specific product. These terms are added to the core of the virtual optimal review. REVRANK then uses those terms to define a feature vector representation of the optimal review. Reviews are then converted to this representation and ranked according to their distance from a 'virtual core' review vector. Our method is fully unsupervised, avoiding the labor-intensive and error-prone manual training annotations typically used in content ranking tasks. We experimented with Amazon reviews on books from several genres, showing that our system clearly outperforms a baseline imitating the user vote model used by Amazon.

The following section discusses related work. Next we present the details of the algorithm. The evaluation setup and results are given in the fourth section. In the Discussion section we discuss and analyze different aspects of the results. We conclude and present directions for future research in the last section.

Related Work

Broadly taken, reader reviews can be thought of as essays having the reviewed product as their topic. Higgins et al. (2006) identify off-topic student essays based on lexical similarity between essays in a collection of essays supposedly on the same topic. Larkey (1998) used clustering and regression models based on surface features such as average length of essay, average length of word and number of different words, ignoring the content all together. Attali and Burstein (2006) report on the evolution of e-rater, a commercial grading system based on several models that analyze discourse segments, stylistic features, grammar usage, lexical complexity and lexical similarity of essays on the same topic. They also provide a review of the field since its early stages. All of these systems require labor-intensive annotation or a set of essays already graded by human graders.

An optimal helpful review could also be thought of as the best summary of many other reviews, each contributing some insight to the optimal review. This is a type of multi-document summarization. In this sense, review ranking is similar to the evaluation of multi-document summarization systems. ROUGE (Lin and Hovy 2003) and Pyramid (Nenkova and Passonneau 2004) are the main methods for evaluation of summarization systems. However, both suffer from the annotation bottleneck, as human-produced summaries (and/or annotations) are required. Our REVRANK needs no annotation at all since we use the vast number of online reviews as the human produced summaries. REVRANK is not a summarization system nor a summaries evaluation system per se although it lies between the two. There has been some work on review summarization: (Hu and Liu 2004) extract product features and output a sentiment-based summary-like list of product features and sentences that describe them. Popescu and Etzioni (2005) improve upon (Hu and Liu 2004).

From a different angle, product reviews are opinions (sentiments) on the product stated from various perspectives. The different perspectives expressed in documents were distinguished based on their statistical distribution divergence in (Lin and Hauptmann 2006).

Wiebe et al. (2004) learn to identify opinionated documents by assigning a subjectivity score (learned from an annotated corpus) to each document. Pang et al. (2002) compare different machine learning algorithms for sentiment classification of movie reviews. Turney (2002) and Dave et al. (2003) classify reviews according to the polarity of the sentiment expressed. We are interested in helpfulness of the review as a whole and not only in what and how strong the sentiment expressed in the review is.

Some studies learn review quality. Liu et al. (2007) prepared and used a large manually annotated training set for identifying low quality reviews of electronic products in a supervised manner, in order to improve the summarization of sentiment regarding product features. Kim et al. (2006) predict the helpfulness of a review by structural features such as length, lexical features, and meta-data such as rating summary (star rating at amazon.com). They use product features extracted from pro/con listings on epinions.com, and sentiment words from publicly available lists. Review subjectivity (where "the reviewer gives a very personal description of the product") was used to predict helpfulness in (Ghose and Ipeirotis 2007).

We differ from the works above in three main aspects. First, our method is fully unsupervised, requiring no manually annotated training set. Liu et al. (2007) employed four annotators, each annotator following very detailed instructions in the course of annotating thousands of reviews for training and evaluation. This is a serious project reporting good results. However, having thousands of reviews annotated and ranked by one evaluator might be problematic, since after a few dozen reviews it is hard for the evaluator to assess the true helpfulness of a review due to heavy interference of information from previous reviews. We also avoid the use of pre-made lists of sentiment words. While (Popescu and Etzioni 2005) use parsers, NER systems and POS taggers in a preprocessing stage, REVRANK does not need them. Avoiding such preprocessing systems increases results quality, since these systems are usually trained on well-written corpora, and thus tend to perform poorly on freely-formed user generated content such as book reviews.

A second difference is that while the works above address electronic products, we focus on book reviews. Whereas electronic products have a relatively small number of features discussed in reviews (typically found in semi-structured specification sheets), book reviewers tend to express themselves more poetically and to discuss many aspects of authoring, such as author style, genre, plot, moral aspects of the story, existential feelings of the characters, and ideas communicated by the author. Not only are these much harder to extract, they are hard to define. We find a flexible set of key concepts, not only those specifically mentioned in the product specs and in pro/con lists.

A third difference from some of the works above is in the evaluation method. While (Kim et al. 2006) measure success by correlation with the Amazon user votes-based ranking, we follow (Liu et al. 2007) in viewing this method as biased. We employed several human evaluators and show that our system outperforms the user-vote baseline.

Finally, keyphrase extraction is another task relevant to our work. (Mihalcea and Tarau 2004) propose the TextRank a graph-based ranking algorithm that ranks keyphrases in an unsupervised way according to concurrences links between words. (Wan and Xiao 2008) propose CollabRank, a similar graph-based ranking model, using the collaborative knowledge given in multiple documents. Like Wan and Xiao (2008), REVRANK exploits a collection of documents (all reviews for a given book) in order create the lexicon of the dominant concepts. REVRANK can tolerate some noise, hence the lexicon is created in a much simpler way than both TextRank and CollabRank.

The REVRANK Algorithm

Overview

REVRANK is based on a collaborative principle. Given multiple reviews of a product, REVRANK identifies the most important concepts. The challenge lies in finding those concepts that are important but infrequent. The main idea employed by REVRANK is to use the given collection of reviews along with an external balanced corpus in order to define a reference *virtual core (VC)* review. The VC review is not the best possible review on this product, but is, in some sense, the best review that can be extracted or generated from the given collection (hence our usage of the term 'virtual': the collection might not contain a single review that corresponds to the core review and the virtual core review may change with the addition of a single new review to the collection). We do not generate the VC review explicitly; all reviews, including the VC one, are represented as feature vectors. The feature set is the lexicon of *dominant terms* contained in the reviews, so that vector coordinates correspond to the overall set of dominant terms. Reviews are then ranked according to a similarity metric between their vectors and the VC vector.

Our approach is inspired by classic information retrieval, where a document is represented as a bag of words, and each word in each document is assigned a score (typically *tf-idf* (Salton and McGill 1983)) that reflects the word's importance in this document. The document is then represented by a vector whose coordinates correspond to the words it contains, each coordinate having the word's score as its value. Similarity of vectors denotes similarity of documents.

The key novelty in our approach is in showing how to define, compute and use a virtual core review to address the review ranking problem.

Our features (dominant terms) constitute a compact lexicon containing the key concepts relevant to the reviews of a specific product. The lexicon typically contains concepts of various semantic types: direct references to the book and the plot, references to similar books or to other books by the same author, and other important contextual aspects. We identify this lexicon in an unsupervised and efficient manner, using a measure motivated by *tf-idf* with the use of an external balanced reference corpus.

There are three main differences from the classic *tf-idf*. First, at the stage of key concepts extraction, we view the collection of reviews for a specific book as a single long document. Second, instead of computing inverted frequencies on the reviews corpus, we use an external balanced reference corpus. Finally, instead of counting the documents in the inverted corpus, we compute the *term*-frequency in the external balanced corpus. The rationale behind this is that we first want to identify the set of key concepts in reviews of a specific book, instead of directly retrieving a single document/review.

Using standard *tf-idf* by viewing each review as a document, we would perhaps be able to discover important terms that appear in very few reviews, but we might not be able to discover central terms that are common to several reviews (since the *idf* tends to punish terms that appear in more than one review). Reviews that make use of such terms are expected to be more helpful than reviews that do not, because in a sense the existence of such terms guarantees that the review also covers the main aspects of the subject at hand.

The REVRANK ranking algorithm has three simple stages, described in detail below: (i) identify dominant terms; (ii) map each review to a feature vector defined by the dominant terms; and (iii) use some similarity metric to rank the reviews.

The Virtual Core Review

Lexical items (one word or more) are associated with the virtual core review according to their *dominance*. As with *tf-idf*, dominant terms are not necessarily the most frequent terms in the reviews collection. The dominant concepts usually cover a wide range of semantic types (to clarify, we do not determine these types explicitly here).

Important types include words common to product reviews in general, such as 'excellent'; words common to reviews of certain types of products (e.g., books and movies), such as 'boring' and 'masterpiece'; words common to reviews of books of certain genres, such as 'thriller' and 'nonfiction'; book specific terms such as names of main charac-

ters and concepts related to the plot (e.g., 'Langdon', 'albino' and 'the holy grail' for the Da Vinci Code). There are also references to other relevant concepts, such as 'Foucault's Pendulum', 'Angles and Demons', 'Christianity', 'historical', 'fictional' (again, for the Da Vinci Code). Explicit handling of these semantic types will be addressed in future papers.

In order to identify the dominant terms, we use a balanced corpus B of general English (we used the British National Corpus (BNC)). This resource is not specific to our problem and does not require any manual effort. The key concepts are identified in the following manner. First we compute the frequency of all terms in the reviews collection. Each term is scored by its frequency, hence frequent terms are considered more dominant than others (stopwords are obviously ignored). Then, the terms are re-ranked by their frequency in the reference corpus B. This second stage allows us to identify the concepts that serve as key concepts with respect to the specific book. For example, the term 'book' or 'Dan Brown', the name of the author of the Da Vinci Code are usually very frequent in the reviews corpus, however their contribution to the helpfulness of a review is limited as they do not provide the potential reader with any new information or any new insights beyond the most trivial. On the other hand, concepts like 'Fuocault's Pendulum' and 'the Council of Nicea' are not as frequent but are potentially important, therefore the scoring algorithm should allow them to gain a dominance score.

Given a corpus R_p consisting of reviews for product p, these two stages can be collapsed so the dominance of a lexical item t in R_p is given by:

$$D_{R_p}(t) = f_{R_p}(t) \cdot c \cdot \frac{1}{\log B(t)} \qquad (1)$$

where $f_{R_p}(t)$ is the frequency of term t in R_p, $B(t)$ is the average number of times t appears per one million words in the balanced corpus (BNC in our case), and the constant c is a factor used to control the level of dominance[1]. The lexicon is constructed ignoring stopwords.

Equation 1 is designed to capture the key concepts that are dominant in the collection R_p, balancing the bias induced by frequent words, capturing words of different semantic types and capturing words that are dominant but infrequent.

Once each term has a dominance score, we choose the *m* most dominant lexical items to create a compact virtual core review. Based on a small development set (reviews for only two books not participating in the test), we used $m = 200$. Clearly, *m* could be determined dynamically to achieve optimal results for different products, but we leave this for future research. In principle, lexical items may have different dominance weights (e.g., given by their score or by their semantic type), allowing several specific ranking schemes. In

---

[1] It should be tuned according to the balanced corpus used. In our experiments we used $c = 3$, so that terms with frequency lower than 0.1% in the BNC will have $c \cdot \frac{1}{\log BNC(t)} > 1$. This way, when using 1 as a dominance threshold, less terms are considered dominant as c increases.

this paper, we set all weights to be 1, which gives excellent results (see the 'Evaluation Setup and Results' Section).

Review Representation and the Virtual Core Review

The compact lexicon of the m most dominant concepts is now used for defining a feature vector representation of reviews. Coordinates of the vector correspond to the dominant concepts. Each review $r \in R_p$ is mapped to v_r in this representation such that a coordinate k is 1 or 0 depending on whether or not the review r contains the kth dominant term.

We refer to the feature vector having 1 in all of its coordinates as the *virtual core* feature vector (VCFV). All reviews that make use of all of the dominant terms are represented by VCFV. In practice, it is rare for a review to use all of the dominant terms, which is why we call it 'virtual'.

Note that VCFV only represents the set of 'core' concepts in the given review collection, and may dynamically change as more reviews are added to the collection.

Again, in both the VCFV and all of the v_r's weights are all set to 1, which could be extended in the future.

Review Ranking

A good book review presents the potential reader with enough information without being verbose. Therefore each review r is given a score by the following equation:

$$S(r) = \frac{1}{p(|r|)} \cdot \frac{d_r}{|r|} \tag{2}$$

where $d_r = v_r \cdot VCFV$ is the dot product of the (weighted) representation vector of review r and VCFV, $|r|$ is the number of words in the review and $p(|r|)$ is a punishment factor given by:

$$p(|r|) = \{{}^c_1 \quad {}^{|r| < |\bar{r}|}_{otherwise} \tag{3}$$

The punishment factor is needed in order to punish reviews that are too short or too long. We set $c = 20$, to deliberately create a high threshold for reviews that are too short. This arbitrary decision was based on our familiarity with Amazon book reviews; the function *p()* could be adjusted to punish or favor long reviews according to user preferences (see Equation 5) or to corpus specific (or forum specific) characteristics. In our experiments we assumed that users tend to get annoyed by verbose reviews; the punishment for an excessive length is already given by the denominator $|r|$ (Equation 2).

Evaluation Setup and Results

The Nature of Book Review Evaluation

The evaluation of tasks such as review ranking is known to be problematic. The main issue is the need for human assessors in order to evaluate tasks. Not only is the definition of a good output vague, success in these tasks is highly subjective. One way to overcome these issue is by having more than one evaluator, then using measures for inter-evaluator agreement in order to estimate evaluation reliability (Liu et al. 2007).

The evaluation of review ranking challenges us with some other issues that are worth mentioning. A good product review is one that helps the user in making a decision on whether to buy or use the product in question. In that sense, the evaluation of reviews is even more problematic. A potential buyer is really interested in the product and we may assume that the way she approaches the immense number of available reviews is mentally different from the mental approach of an objective evaluator who has no real interest in the product. We refer to this problem as the *motivation* issue. For example, it might be the case that a real buyer will find interest in a poor review if it adds to the information he already has.

Finally, the evaluation procedure of this type of tasks is rather expensive. We propose an evaluation procedure that balances these constraints.

There are two main approaches to deal with review ranking, none of which fully overcomes the motivation issue. The first approach, as used in (Liu et al. 2007), is to define strict and detailed guidelines and have the human annotators annotate and evaluate the reviews according to those guidelines. Liu et al. (2007) used 4 annotators, each annotating and evaluating thousands of reviews. Given a review, their annotators were asked to indicate whether a decision could be made based on this review solely, whether the review is good but more info is needed, whether the review conveys some useful piece of information although shallow, or whether the review is simply useless. Due to the usage of strict rules, this method is a good one. However, the evaluation is still subjective and the motivation issue is still present. In addition, using a small number of annotators supposedly ensures consistency, but could also result in a kind of 'early bird' bias, since annotators become product experts when reading so many reviews. For these reasons, this methodology is more suitable to products such as electronic products, which have a relatively small number of well-defined features, than to books, movies and music, on which opinions can be highly open ended.

In the other approach, instead of having a small number of evaluators evaluate a very large number of reviews according to strict guidelines, we could have a large number of evaluators, each evaluating a small number of reviews according to loosely-defined guidelines. While more expensive, this approach is closer to amazon's voting system, in which many potential buyers read a relatively small number of reviews and indicate whether they are useful or not. When applied to book reviews, this method is more suitable to deal with the motivation issue, since the random evaluator who is not actively interested in getting the product relates to book reviews much easier than to the technical details and lingo of the review of some electronic gadget.

Both methods have their pros and cons. We chose the second, both because it is closer to the real conditions at which ranking reviews according to helpfulness is needed and because it is more suitable to the type of products we wanted to address (books).

Evaluation Procedure

Obviously, the forum a review is posted at influences the expectations of the users (its potential readers). For this research (for our human subjects) we define a *helpful* review in a somewhat loose manner as follows[2]: *A good (helpful) book review presents the potential reader with sufficient information regarding the book's topic/plot/writing style/context or any other criteria that might help him/her in making a rational decision on whether or not s/he wants to read or buy the book.*

We compared user votes-based ranking to our REVRANK ranking, and assessed performance by asking human evaluators to rate the reviews. User votes-based ranking takes two factors into account: the total number of votes a review receives, and the ratio between the number of helpful votes and the total number of votes. Both factors are needed since a review that has a single 'helpful' vote (100% of the votes are positive) cannot be automatically considered as more helpful than a review with helpful/total ratio of 281/301. We tried to approximate the user votes based-ranking employed by Amazon, ranking reviews that have more than 15 votes such that their helpful/total ratio is over 0.5 higher than reviews with a slightly better ratio but under 15 total votes. In a similar way, we ranked reviews with more than 30 votes and a majority of positive votes above those with less than 30 votes. We do not claim that this is the exact algorithm used by Amazon, but we verified (on hundreds of reviews) that it is close enough for our purposes.

We tested our system on five books from five different genres, using the Amazon API to obtain the reviews. The books are: *The Da Vinci Code*, historical fiction (by Dan Brown); *the World is Flat*, non-fiction (by Thomas Friedman); *Harry Potter and the Order of the Phoenix*, fantasy (by JK Rawlings); *Ender's Game*, science fiction (by Orson Scott Card), and *A Thousand Splendid Suns*, fiction (by Khaled Hosseini). All five books are worldwide bestsellers and enjoy many Amazon reviews. Table 1 presents some details about the reviews of these books. The table shows that the books differ in almost every aspect: genre, number of reviews, average review length, and average number of votes.

The purpose of our evaluation is three-fold. First, we verify that user votes helpfulness ranking is biased. Second, we demonstrate that ranking based on REVRANK succeeds in finding helpful reviews among Amazon reviews that received no user votes at all. Finally, we show that REVRANK ranking is preferable even when comparing to the top 2% reviews ranked according to user votes.

The evaluation process is as follows. For each of the five books, we created four batches of six reviews each. Each batch was compiled as follows: two reviews were randomly sampled from the top 2% or 5% of the user votes-based re-

Book	NoR	Length	Votes	Dom
DVC	3481	175.3	15.5	11.1
WiF	1025	195.4	14.1	14.4
HP	5000	182.8	3.4	17.3
EG	2433	146.5	2.6	15.9
TSS	784	120	4.6	14.8

Table 1: Corpus Data. NoR: Number of Reviews. Length: average number of words in a review. Votes: the average number of votes (both positive and negative votes) given for a review of the book by Amazon users. Dom: the average number of dominant lexical features found by REVRANK in a review. DVC: The Da Vinci Code; WiF: The World is Flat; HP: Harry Potter and the Order of the Phoenix; EG: Ender's Game; TSS: A Thousand Splendid Suns.

Book	IE Agreement	kappa, $d = 2$	kappa, $d = 1$
DVC	75%	0.75	0.57
WiF	79%	0.79	0.65
HP5	83%	0.69	0.5
EG	71%	0.71	0.46
TSS	58%	0.64	0.57
Overall	73.3%	0.69	0.56

Table 2: Percentage of inter-evaluator agreement for $d = 2$ and Fleiss' kappa statistics for $d = 1, 2$. The kappa statistic shows significant agreement for both ds. For 73.3% of the total of 120 evaluated reviews, there was complete agreement between three evaluators.

views[3], two reviews were randomly sampled from the top 2 (or 5)% of reviews as ordered by our REVRANK system, and two reviews (control set) were randomly sampled from the reviews that are not included in the top 2 (or 5)% reviews of the user votes-based ranking (some of them could be very helpful, but they had not accumulated enough votes).

Each batch of six reviews was given to three human evaluators to rank the reviews according to helpfulness, assigning 6 to the most helpful review and 1 to the least helpful review.

Overall, we sampled 24 reviews for each book, 120 reviews in total. Each review was evaluated by three different evaluators, having 360 evaluations. Note that each evaluator was given only one batch of 6 reviews for a book, in order to prevent him/her from being cognitively loaded with information in a way that might interfere with his/her judgment.

Inter-evaluator Agreement

Review evaluation is subjective. However, a strong inter-evaluator agreement observed for a number of evaluators evaluating each set of reviews can indicate that one system is superior to another (see (Liu et al. 2007) regarding review ranking and the extensive literature regarding the evaluation of summarization and question answering systems). Inter-evaluator agreement in this evaluation task is somewhat

---

[2]This definition also complies with Amazon's review writing tips ("What makes a good review?"), indicating: "be detailed and specific. What would you have wanted to know before you purchased the product?". Amazon's guidelines are presented to review writers, and ours to our evaluators.

[3]To reduce noise, our preference was to use the top 5%. However, in three of the books in our experiments, only the top 2% reviews had more than 15 votes each with a helpfulness ratio higher than 0.5.

Book	System	Length	Votes	Dom
DVC	UV	308.6	103.8	16.7
	RVR	279.1	51.4	24.7
WiF	UV	335.5	94.4	14.1
	RVR	334.4	10.6	25.6
HP	UV	345.4	51.2	27.7
	RVR	211.8	1.7	30.7
EG	UV	379.5	60.52	31.3
	RVR	188.9	2.3	33.1
TSS	UV	400.8	46.5	34.6
	RVR	142.7	3.5	24.5
All books	UV	353.8	71.3	124.4
	RVR	202.8	13.8	27.7

Table 3: Averages over the top reviews of the user votes-based ranking (UV) and our REVRANK system. Votes: number of votes (positive and negative). Dom: dominant lexical features.

tricky. Each evaluator is asked to assign each review with a number between 1 and 6. Reviews differ in style and coverage. Evaluators differ in their expectations and tolerance for different review styles (one evaluator indicated she prefers cynical reviews while other explained that a certain review is 'too cynical'). Given the fact that the range of evaluation scores was 1 to 6, and given the different personal preferences of the evaluators and the open style of review writing, it is beyond our expectation that evaluators will show strict uniformity. It is therefore reasonable to define a softer measure of agreement. We define agreement as follows. Denote by e_i the three evaluators of review r, and by *score* the function $score(e, r) : \{e_1, e_3, e_3\} \times \{r \in R\} \longrightarrow \{1, ..., 6\}$. Then $agr(e_1, e_2, e_3) := 1$ iff

$$\forall i \neg \exists j \quad score(e_i, r) - score(e_j, r) > d \quad (4)$$

Otherwise, $agr(e_1, e_2, e_3) = 0$. Since scores vary from 1 to 6, we set $d = 2$, assuming that if all 3 evaluators scored r within 2 scores distance we can say there is an agreement.

Fleiss' kappa is a common statistic that measures agreement between multiple evaluators. Table 2 shows the ratio of the agreement and the Fleiss' kappa statistic for $d = 2$ and $d = 1$, demonstrating substantial agreement for $d = 2$ and quite a good agreement for $d = 1$.

Results

Figure 1 presents the performance of our system compared to the user votes ranking and to a random sample. Our system significantly outperforms both of these methods. Note that the maximal grade is 5.5, not 6, because every batch contained *two* reviews selected by each competing system, so the best a system can do is win the first two places with grades of 6 and 5. Table 3 shows that the average number of votes for the top reviews ranked by REVRANK is only 13.8, while the top user-based reviews have an average of 71.3 votes. This shows that our system indeed recognizes worthy reviews overlooked by users.

As can be viewed in Table 3, another interesting observation is the large difference in length between the reviews
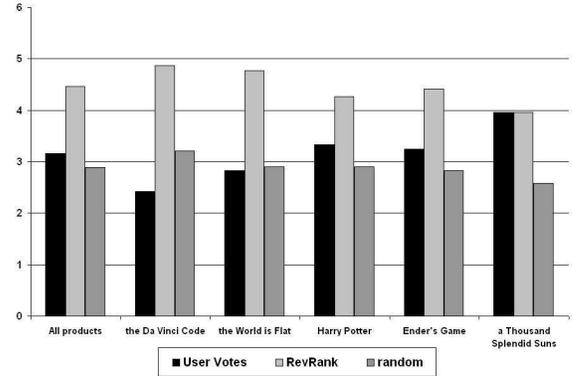

Figure 1: Average evaluation score for the REVRANK system, Amazon user vote-based baseline, and a random sample. Note that the optimal score for a ranking system can only be 5.5, in case it wins the top two places (5 and 6). Similarly, the lowest possible score is 1.5.

chosen by the user votes (UV) based algorithm and the reviews chosen by REVRANK, with an average length of 358.8 for the UV and only 202.8 for REVRANK. Looking at both the average length and the average number of dominant terms found in the reviews shows that REVRANK reviews convey much more information in much shorter reviews. Our results clearly show that users (human evaluators) prefer these compact reviews. This observation will be discussed in greater detail in the Discussion section.

Examining the evaluators' choice of a single most helpful review in each batch shows that an REVRANK *choice was ranked the most helpful in 85% of the batches*.

Comparing our subjects' appraisal of user-based ranking to the control set shows that the former is only slightly better than the latter. We attribute this to the fact that the random sample may contain quality reviews that have not accumulated enough votes, therefore were not part of the UV sample space. These results, along with the difference in the average number of votes (Table 3), confirm that user-based ranking could be problematic.

Discussion

Although our system performs significantly better than the user vote-based ranking system, there is one book (*A Thousand Splendid Suns*) on which both are evaluated similarly. We note that even a similarity in the evaluation score proves the benefits of REVRANK since the system (a) is unsupervised, and (b) finds reviews otherwise overlooked by the users. This result can be explained by differences in average review length. Table 1 shows that while the average length of reviews for the first four books is 175 words per review, the average length for *A Thousand Splendid Suns* is only 120 words. The REVRANK ranking algorithm balances the length of the review and the number of pieces of information it conveys, and as such, it favors shorter reviews relative to the average length of the reviews for the book in question (see Equation 2). Apparently, human evaluators

Flattening Flatteners are flattening the world...
I listened to Tom Friedman's book on CD, and this is what it sounded like: "I realized the world was flat when the flattening flatterers converged and flattened the world into flatness. Flatteners one through ten flattened the world but when they converged into convergence with a TCP/IP SOAP XML in-source outsource delivery flattening, the flatness became overriddingly flat and converged..."
In fairness, the book is a good piece of scholarship and shoe leather, and Friedman does help educate a good portion of the world on how all these forces intertwine. But hearing the book on CD makes you see how Friedman hammers this stuff over and over again like a two-by-four to the middle of your forehead.

Table 4: Excerpt of review #664 for *The World is Flat*.

(and readers) tend to prefer reviews that are longer than a certain constant, 10 sentences or 150 words approximately[4]. This can be easily supported by REVRANK by adjusting the punishment function (Equation 3) to be:

$$p(|r|) = \begin{cases} c & |r|<|\overline{r}| \quad or \quad |r|<l \\ 1 & otherwise \end{cases} \quad (5)$$

where l is a constant determining a lower bound on the length of a helpful review. Equation 5 could be also used to fit the preferences of a specific user who prefers shorter or longer reviews.

We note that the hypothesis about the length preference of the user is supported by our results, however we are not aware of any cognitive or computational work that specifies a length of preference. Length preference is subjective and may vary between domains. We leave tuning of length preferences to a future study.

While the reviews selected by REVRANK obtained an average rank of 4.5, review 664 for *The World is Flat*, sampled for the 6th batch, has an average evaluator rank of 2 (two of the three evaluators ranked it as the least helpful review in the batch). Looking at the review itself reveals that this review is a parody, criticizing the book's extensive and repetitive use of buzz words (see Table 4). The reviewer used all the right terms but deliberately misplaced them in order to form sentences that are extremely incoherent, hence not presenting the reader with any explicit information about the book. This review was ranked relatively high by REVRANK due to the fact that the system sees the reviews as a bag of words, and is thus insensitive to the order of the words. It is also worth noting that this review got 15 votes from Amazon users, 13 of which were 'helpful'. These votes make a helpfulness ratio of 87%. The review did not obtain such a high helpful score by the user-based baseline because it did not accumulate enough votes (over 30), but apparently many readers liked the parody and thought that the message it conveys is helpful.

Another aspect worth mentioning is that REVRANK is best viewed as a tool for finding the top n reviews and not the *one* best review. This is due to the facts that (a) there is

probably no *one* best review, (b) review evaluation is after all subjective, and (c) reviews can complement each other, providing an added value to the potential reader. Users typically read more than the very top review for a product. The algorithm could be easily configured to output the top n orthogonal reviews that have the least overlap of key concepts. This will present the user with a set of complementary reviews. Doing so directs the effort toward multi-review summarization, a good subject for future work.

Conclusion

Book reviews greatly vary in style and can discuss a book from very many perspectives and in many different contexts. We presented the REVRANK algorithm for information mining from book reviews. REVRANK identifies a compact lexicon (the virtual core review) that captures the most prominent features of a review along with rare but significant features. The algorithm is robust and performs well on different genres. The algorithm is fully unsupervised, avoiding the annotation bottleneck typically encountered in similar tasks. The simplicity of REVRANK enables easy understanding of the output and an effortless parameter configuration to match the personal preferences of different users.

For future work we plan to experiment with even smaller review sets. In addition, we will study the effects of different weighting schemes in order to enable a personalized ranking according to various criteria. Finding the optimal set of orthogonal reviews will also be studied. We will eventually generate one comprehensive review from the most valuable orthogonal reviews chosen by our ranking system. The application of the algorithm on the blogosphere as well as other social media domains can also be investigated, for creating a vertical search engine that identifies the most helpful reviews on the web and not only on a restricted corpus. Finally, the expensive nature of the evaluation process raises an essential direction of future work: developing a gold standard for evaluation purposes.

References

Attali, Y., and Burstein, J. 2006. Automated essay scoring with e-rater v.2. *The Journal of Technology, Learning and Assesment* 4(3).

Dave, K.; Lawrence, S.; and Pennock, D. M. 2003. Mining the peanut gallery: opinion extraction and semantic classification of product reviews. In *WWW '03: Proceedings of the 12th international conference on World Wide Web*, 519–528. New York, NY, USA: ACM.

Ghose, A., and Ipeirotis, P. G. 2007. Designing novel review ranking systems: predicting the usefulness and impact of reviews. In *ICEC '07: Proceedings of the ninth international conference on Electronic commerce*, 303–310. New York, NY, USA: ACM.

---

[4]Amazon review writing tips suggest "Not too short and not too long. Aim for between 75 and 300 words".

Higgins, D.; Burstein, J.; and Attali, Y. 2006. Identifying off-topic student essays without topic-specific training data. *Nat. Lang. Eng.* 12(2):145–159.

Hu, M., and Liu, B. 2004. Mining and summarizing customer reviews. In *KDD '04: Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, 168–177. New York, NY, USA: ACM.

Kim, S.-M.; Pantel, P.; Chklovski, T.; and Pennacchiotti, M. 2006. Automatically assessing review helpfulness. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 423–430.

Larkey, L. S. 1998. Automatic essay grading using text categorization techniques. In *Proceedings of SIGIR-98, 21st ACM International Conference on Research and Development in Information Retrieval*, 90–95. Melbourne, AU: ACM Press, New York, US.

Lin, W.-H., and Hauptmann, A. 2006. Are these documents written from different perspectives?: a test of different perspectives based on statistical distribution divergence. In *ACL '06: Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the ACL*, 1057–1064. Morristown, NJ, USA: Association for Computational Linguistics.

Lin, C.-Y., and Hovy, E. 2003. Automatic evaluation of summaries using n-gram co-occurrence statistics. In *NAACL '03: Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, 71–78. Morristown, NJ, USA: Association for Computational Linguistics.

Liu, J.; Cao, Y.; Lin, C.-Y.; Huang, Y.; and Zhou, M. 2007. Low-quality product review detection in opinion summarization. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, 334–342.

Mihalcea, R., and Tarau, P. 2004. Textrank: Bringing order into texts. In *Conference on Empirical Methods in Natural Language Processing*.

Nenkova, A., and Passonneau, R. J. 2004. Evaluating content selection in summarization: The pyramid method. In *HLT-NAACL*, 145–152.

Pang, B.; Lee, L.; and Vaithyanathan, S. 2002. Thumbs up? Sentiment classification using machine learning techniques. In *Proceedings of the 2002 Conference on Empirical Methods in NaturalLanguage Processing (EMNLP)*, 79–86.

Popescu, A.-M., and Etzioni, O. 2005. Extracting product features and opinions from reviews. In *HLT '05: Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, 339–346. Morristown, NJ, USA: Association for Computational Linguistics.

Salton, G., and McGill, M. J. 1983. *Introduction to Modern Information Retrieval*. McGraw-Hill New York, NY.

Turney, P. D. 2002. Thumbs up or thumbs down? semantic orientation applied to unsupervised classification of reviews. In *ACL '02: Proceedings the 40th annual meeting of the ACL*, volume 40, 417.

Wan, X., and Xiao, J. 2008. Collabrank: Towards a collaborative approach to single-document keyphrase extraction. In *Proceedings of the 22nd International Conference on Computational Linguistics (COLING)*.

Wiebe, J.; Wilson, T.; Bruce, R.; Bell, M.; and Martin, M. 2004. Learning subjective language. *Computational Linguistics* 30(3):277–308.