Graduate School ETD Form 9
(Revised 12/07)

PURDUE UNIVERSITY
GRADUATE SCHOOL
Thesis/Dissertation Acceptance

This is to certify that the thesis/dissertation prepared

By Eric Lin

Entitled
Text mining of online book reviews for non-trivial clustering of books and users

For the degree of Master of Science

Is approved by the final examining committee:

Shiaofen Fang
Chair

Snehasis Mukhopadhyay

Yingzi Du

To the best of my knowledge and as understood by the student in the *Research Integrity and Copyright Disclaimer (Graduate School Form 20)*, this thesis/dissertation adheres to the provisions of Purdue University's "Policy on Integrity in Research" and the use of copyrighted material.

Approved by Major Professor(s): Shiaofen Fang

Approved by: Shiaofen Fang 5/14/12
Head of the Graduate Program Date

Graduate School Form 20
(Revised 9/10)

PURDUE UNIVERSITY
GRADUATE SCHOOL

Research Integrity and Copyright Disclaimer

Title of Thesis/Dissertation:

Text mining of online book reviews for non-trivial clustering of books and users

For the degree of Master of Science

I certify that in the preparation of this thesis, I have observed the provisions of *Purdue University Executive Memorandum No. C-22,* September 6, 1991, *Policy on Integrity in Research.**

Further, I certify that this work is free of plagiarism and all materials appearing in this thesis/dissertation have been properly quoted and attributed.

I certify that all copyrighted material incorporated into this thesis/dissertation is in compliance with the United States' copyright law and that I have received written permission from the copyright owners for my use of their work, which is beyond the scope of the law. I agree to indemnify and save harmless Purdue University from any and all claims that may be asserted or that may arise from any copyright violation.

Eric Lin
_____
Printed Name and Signature of Candidate

5/14/12
_____
Date (month/day/year)

*Located at http://www.purdue.edu/policies/pages/teach_res_outreach/c_22.html

TEXT MINING OF ONLINE BOOK REVIEWS FOR NON-TRIVIAL CLUSTERING OF BOOKS AND USERS

A Thesis

Submitted to the Faculty

of

Purdue University

by

Eric Lin

In Partial Fulfillment of the

Requirements for the Degree

of

Master of Science

August 2012

Purdue University

Indianapolis, Indiana

To my parents, without whom I would not be possible…

ACKNOWLEDGEMENTS

There are many people I would like to thank, who have made this project possible. First, I would like to thank Dr. Shiaofen Fang, my advisor. His guidance has been invaluable to me throughout this project.

I would also like to thank Dr. Snehasis Mukhopadhyay and Dr. Eliza Yingzi Du for agreeing to serve on my thesis committee. Dr. Yuni Xia also deserves a mention, for her feedback in the early stages of this project.

Though I do not know them personally, I would like to thank the Goodreads community who wrote the reviews I used in this thesis, as well as the team at Goodreads, for providing me with access to their data.

Finally, I would like to thank my family, for all of their love and support.

TABLE OF CONTENTS

Page

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

Lin, Eric. M.S., Purdue University, August, 2012. Text Mining of Online Book Reviews for Non-trivial Clustering of Books and Users. Major Professor: Shiaofen Fang.

The classification of consumable media by mining relevant text for their identifying features is a subjective process. Previous attempts to perform this type of feature mining have generally been limited in scope due having limited access to user data. Many of these studies used human domain knowledge to evaluate the accuracy of features extracted using these methods. In this thesis, we mine book review text to identify nontrivial features of a set of similar books. We make comparisons between books by looking for books that share characteristics, ultimately performing clustering on the books in our data set. We use the same mining process to identify a corresponding set of characteristics in users. Finally, we evaluate the quality of our methods by examining the correlation between our similarity metric, and user ratings.

CHAPTER 1. INTRODUCTION

In 2009, 288,355 books were published by print, a drop of half a percent from the year before. By comparison, 764,448 titles were published using other methods, representing an increase of 181% from 2008. Despite traditional book publishers declining as a player in the book market, the total number of books published annually has actually increased year by year, largely due to the increasing number of books that have been self-published, or other nontraditional means. Unlike other forms of consumable media (music, movies, television), which have prohibitively high production costs, the cost to publish a book is extremely low. In addition, the electronic book format has greatly reduced authors' dependence on book publishers as the primary means of book distribution, contributing to the steady increase in total book production: in 2008, the total number of books produced broke one million units for the first time [1].

As the number of new books being published every year increases, the decision involved in picking a new book to read becomes more difficult as well, a paradox of choice effected by this flood of options. This process of book discovery is one of the biggest problems that readers face today.

Although the opinions of friends remain the most common (and trusted) method of book discovery, these are limited in two ways. First, the recommender is only capable of recommending books they have already read, and secondly, the recommender may not have a complete understanding of the type of book the reader is interested in reading. Given these limitations, book discovery can be an extremely challenging problem to solve.

Goodreads [2] is a social network for readers, created in 2006. On Goodreads, users are able to maintain a catalog of books they have read, including their overall opinion of the book, expressed in a 5-star rating, and more detailed thoughts about the book, in the form of written reviews.

In the current age of information, information is being generated and collected at a higher rate than ever before. We believed that existing data mining methods could be used to identify clusters of similar books, using the treasure trove of review data collected from the users on Goodreads.

To date, Goodreads has over nine million registered users, who added a total of 320 million book ratings to the Goodreads database. This database of users and their review data provided us with an enormous set of book reviews for text mining, and a way for us to make connections between books and users, by associating a book review with the user who wrote it. This association allowed us to get a more complete picture of the users who wrote each review. Through the

data available in the Goodreads database, we were able to see what other books that user has read, how highly they rated each of those books, and use this information to inform an analysis of user rating habits.

The quality of the Goodreads data allowed us to tackle the problem of book discovery in a unique way. We believed that by mining the aggregate of a book's review text, we would be able to identify key characteristics present in that book. By performing this mining process for multiple groups, we hoped to be able to categorize groups into naturally forming clusters based on the characteristics that can be mined from their review text.

Books can be grouped in many ways. The most obvious groupings are based on objective classifications: it is fairly simple to determine if a book is a historical autobiography, or American literature from the Great Depression. Though these distinctions can be useful, we consider them to be *trivial* classifications, because such distinctions are obvious, concrete, and generally agreed-upon. They are distinctions that can be made quite easily without the use of text mining. The real challenge lay in classifying books using less-obvious identifiers. These characteristics, which we referred to as *nontrivial* attributes, are less obvious characteristics, which play a large part in determining a book's identity, but are difficult to identify. An author's tone, the style of narrative, or the social commentaries embedded in a book's story are all examples of nontrivial

attributes. Moreover, these nontrivial attributes can be combined with each other, or trivial attributes to define extremely nuanced subsets of books.

In this study, we propose the use of text mining to classify books into nontrivial clusters using book review data from Goodreads with Bookmine, a tool we developed for this purpose. We intended to accomplish this goal by identifying frequently occurring 'feature' tag words, and grouping books according to the extent which these traits were expressed in a book's reviews. Our underlying assumption was that a book's review text contained descriptions of a book's characteristics. By mining this text, we expected to be able to identify the book's defining characteristics. Furthermore, we expected similar books to have similar attributes present in their review text. It was our hope that by clustering books by the commonalities among the characteristics mined from their reviews, we would be able to identify groups of books that are similar in meaningful, nontrivial ways.

Since the goal of this project was the formation of nontrivial book clusters, we were careful when making decisions about the books that would be mined. We were concerned that mining a data set containing books from too many different genres would cause genre-specific features to overwhelm other features, diluting the impact of nontrivial attributes. To avoid this case, we limited our data set to books from within the same genre. We used the books from National Public Radio's list of the top 100 science fiction and fantasy books, published in August of 2011 [3].

CHAPTER 2. RELATED WORK

Mining unstructured text inevitably requires some method to reduce the sheer volume (and often, the dimensionality), of data. Feldman and Dagan performed some of the seminal work on mining keywords from text, and performing analysis on the text using the keywords in comparison operations [6][7]. Most basic automated text mining techniques are variations of the term frequency-inverse document frequency method (TF-IDF) [4][5]. This method of determining the weight of terms found in a document accounts for terms that occur frequently, while simultaneously placing greater importance on terms that occur less frequently.

Newer tools such as WordNet [8] have been used as part of this process, to improve keyword selection through the inclusion of additional measures to assist with the semantic interpretation of the mined texts during this process, whether by allowing similar concepts to be combined, or by organizing ideas into a hierarchical framework.

The process of obtaining keywords as a preliminary step to facilitate textual analysis is usually performed by mining the text for a set of count vectors,

corresponding to the frequency words (or sometimes phrases and ideas) occur in the data. Research with the intent to reduce the dimensionality of these count vectors has suggested that mapping these count vectors to a lower-dimensional space can be beneficial in reducing the impact of noise when mining text [9].

These studies suggest keywords are a valid method of summarizing unstructured data in a meaningful way, and furthermore, that reducing the dimensionality of this data can often have the effect of reducing the impact of noise in the analysis.

In the domain of mining the text of human (user) written reviews, the idea of sentiment analysis, or the interpretation of the human's subjectivity become increasingly important. Some studies have used visualization techniques to assist with the identification and evaluation of identifying keywords [10], and the classification of reviews into emotive (positivity or negativity) categories [11], while others have used visualization to identify trends in the data by visualizing the summarized data directly [12]. Pang and Lee [13][14] discuss many of the issues and challenges that come up when mining human reviews [14].

Most studies of mining a large amount of text focus on finding interesting relational patterns from frequently occurring entities in the data. The distinction between of 'interesting' and 'uninteresting' patterns has been studied in [15][16], though most of these studies do so in the domain of the evaluation of association rules.

The analysis of user reviews has been explored at some length, including an adaptive solution for multiple domains proposed by Blitzer et al [17], and a keyword-based approach to classifying books [18], similar to the method used in this study. In their work, Wanner et al [18] identify books as pertaining to a predetermined set of topics in their sample books, using human opinion to evaluate their topic detection algorithm. Although a correlation was found between topic significance, as determined by their algorithm, some cases were noted where the results of topic detection were misleading. Their results are discussed in more detail in our methodology discussion.

This thesis also draws on work that examines methods to evaluate similarity in text [19], focusing primarily on vector-based approaches. Euclidean distance and cosine angle distance are two of the most widely used methods utilized to quantify similarity (or difference) between texts. Work to make comparisons between the two methods show that they perform similarly at high dimensions, while cosine distance can be advantageous due to the normalized distances produced as a result [19]. Others have built upon these methods, by measuring the semantic similarity between text passages. Mihalcea et al evaluate the semantic similarity between phrase-pairs [20], reporting an improvement over simple lexical matching, though the nature of their study is primarily tailored for evaluating similarity between shorter fragments of text.

With the increasing availability of user data, efforts to identify user interests by sentiment analysis of review data, and the application of these results to make recommendations have received more attention. Over time, as the volume of data has grown by several degrees of magnitude, and as techniques and processing power have improved, there has been a shift from approaches that rely on human interaction as part of the initial identification of feature from content [21][22], to methods that use human interaction as a tool of evaluating the results of algorithm-based methods to produce these results. Others have gone further, asserting that user preferences are not constant, and are in fact dependent on factors such as time and location, proposing methods to take these factors into account when identifying user preferences [23]. Techniques to summarize and categorize data are still largely dependent on human evaluation to generate meaningful results [24], and will likely remain so for the foreseeable future.

Although our primary discussion points in this thesis evaluate the viability of detecting book clusters by mining user reviews, the most likely application of this type of study is in the realm of making generalizations and predictions using the resulting clusters. Most studies, such as those sponsored by the Netflix Prize, are interested in making recommendations based on these generalizations [25][26][27].

When making recommendations through generalization, there are typically two approaches: those based on clustering a user with other users (a clique-based approach), and those based on recommending products with similar features, determined by mining content, or some other means. Alspector et al [28] compare the two approaches in their work, in which users are polled to determine their movie preferences. Their findings demonstrated that clique-based were better suited for capturing user preferences, which tended to be extreme at times. However, this approach is incapable of recommending newer movies, due to a lack of rating data. On the other hand, a feature-based is capable of making recommendations for newer movies, and for selectively targeting users who are interested in specific features, but is dependent on identifying features correctly. The study concludes by recommending a hybrid approach to take advantage of both methods, as is attempted by Campos et al [29].

This thesis attempts to build upon these efforts to form meaningful content-based clusters. We propose an extension of earlier attempts to build content-based clusters of items into the user domain, by mining features from the content of user-written reviews about books in our data set. Furthermore, we propose the formation of a corresponding set of user clusters, by treating each user as an entity defined by the sum of their authored review content. Effectively, we utilize methods of creating content-based clusters to form cliques of users as well. As far as we can determine, the data necessary for this type of dual clustering has not been available in studies involving the book domain. Finally, we evaluate

validity of this method of clustering both books and users by examining the correlation between the two types of clusters, as evidenced by user book ratings.

CHAPTER 3. METHODOLOGY

3.1 Data Collection and Preprocessing

Review data for the 100 books selected for our data set were pulled from the Goodreads database, consisting of user reviews written about each of those books. This data also included user ratings.

Preliminary data preprocessing was performed before mining the review data. Non-English words, and words not contained in a standard dictionary were removed, including misspelled words. Additionally, user identifiers such as a user's real name and email address were removed. It should be noted that Goodreads is an international community of readers, and reviews written by international Goodreads users were removed in this step.

3.2 Mining Content

Each book's reviews were mined for frequently occurring words, producing a set of vectors corresponding to the frequency of each word. This process was performed independently for each book, resulting in a different set of vectors for each word. Frequently occurring words were referred to as candidate tags.

The total incidence of a candidate tag word in a book's aggregated reviews is usually a good indicator of the general relevance of that candidate tag to the book. However, this approach greatly exaggerates the importance of highly occurring (but otherwise meaningless) candidate tags, such as "the", "an", or "book".

To account for the skewed nature of purely incidental tag counts, as well as the varying amounts reviews for each book, it was necessary to perform some sort of normalization. For each word in a book's reviews, its weight was determined using the TF-IDF statistic, named for the two terms multiplied together to produce this measure. TF-IDF is shown in (1). The first term, the term frequency, is the quotient of T_{ik}, the number of occurrences of the word *k* in the reviews of a book, and *N*, the total number of reviews for that book. The second term is known as the inverse document frequency, where n_k is the number of reviews that contain the word. When using TF-IDF, a word's term frequency is multiplied by its inverse document frequency, which equates to a measure of the rarity of a particular word. This causes words that occur very frequently to have their weights diluted somewhat by the IDF, while infrequent words have their weights increased.

$$Weight = \frac{T_{ik}}{N} \times \log\left(\frac{N}{n_k}\right) \quad (1)$$

Using TF-IDF, the weight of the “evil” candidate tag for a book with 100 reviews, and 40 counts of the word “evil”, appearing in a total of 20 reviews would be:

$$Weight_{evil} = \left(\frac{40}{100}\right) \times \log\left(\frac{100}{20}\right) \approx 0.6438$$

After mining the weights of candidate tags for each individual book, the mean weight of each candidate tag was calculated across the entire data set. These were considered to be the ‘global’ weights for each candidate tag. Ultimately, candidate tags with high weights were the pool our eventual feature tags were selected from.

3.3 Selecting Feature Tags

Before selecting candidate tags as feature tags, the candidate tags with the highest global weight values were subjected to human evaluation. This was necessary to remove tag words that were lacking in description, too low in overall frequency, or otherwise unsuitable. Table 1 lists the candidate tags with the highest global weight, as well as the results of the human tag filtering processing.

Table 1 High-weight candidate tags mined by Bookmine

Word	Count	Global Weight	Selected as tag?
book	306391	0.992	N
read	178897	0.613	N
story	98901	0.338	N
really	74574	0.260	N
elric	347	0.257	N
series	34425	0.208	N
science	16970	0.206	Y
fantasy	24247	0.202	Y
reading	53636	0.187	N
think	44516	0.146	N
love	49924	0.143	N

Words such as ‘book’, ‘read’, ‘story’, ‘really’, ‘reading’, ‘think’, and ‘love’ were removed due to their ambiguity: they do little to distinguish features one book may have, that another does not. ‘Elric’ is the name of the titular character in The Elric Saga, by Michael Moorcock, and is subsequently mentioned in a high proportion of reviews written about the series. It also received an extremely high weight, due to the IDF term of TF-IDF. Although this type of candidate tag could be useful for finding books about the same character, and because only one of these books existed in our data set, we felt it was too specific of a candidate tag to be considered a feature. ‘Series’, on the other hand, was a fairly meaningful candidate tag, describing whether or not the book being reviewed was part of a series. While useful, this essentially a trivial classifier, the type of identifier we were trying to avoid. The ‘science’ and ‘fantasy’ tags, while comparably general, were selected because they describe content. Had the data set been restricted

further to include only books from either the science fiction or fantasy genre, they would have been eliminated as candidate tags as well.

We selected thirty tags out of the remaining candidate tags, to be used for the duration of the study, which we referred to as feature tags. These are shown in Table 2. We decided on this number of feature tags because we felt it was the lowest amount of tags that would be able to adequately cover the breadth of book features we felt were present in the books of our data set. As part of the selection process, we combined duplicate tags that overlapped to some degree (the words “politics”, and “political”, for instance). In future work, tools such as WordNet [8] can be employed to combine synonymous tags and concepts more intelligently.

Table 2 Bookmine feature tags, with counts and global weights

Word	Count	Global Weight
science	16970	0.20561306302900387
fantasy	24247	0.20168259951844095
classic	11964	0.1337030609327309
dark	9876	0.09726614116584428
space	4632	0.09356455205636464
epic	6075	0.08840912551124937
magic	7614	0.08778473658702554
adventure	5085	0.08517610571758537
entertaining	5531	0.08050868384161108
evil	6354	0.07934201625561284
modern	5051	0.07254866549958161
political	6653	0.07247767580841079
complex	4143	0.06731480208645405
technology	3222	0.06665863369621694
hero	3637	0.06641755317672293
compelling	4194	0.06062477300340192
alien	2630	0.05988180067791569
deep	3608	0.05978172562877917
simple	3704	0.05958141080877874
social	3773	0.05780310281091399
small	3444	0.05770286256399173
intriguing	3516	0.05585336078858344
reality	4209	0.05527132541715071
religion	3822	0.05456477158013236
exciting	3037	0.05392080172682925
sad	6359	0.05256410668563414
sex	5902	0.05197692599651009
battle	3356	0.05057012744229512
humor	3831	0.050453539433717304
adult	3789	0.04869194762604648

3.4 Book Similarity

The use of feature tags provided a context with which to quantify the content of books, since each book could be described by the collection of its weight counts for each of the feature tags. For each book *b*, the weight of tag word *w* in *b* was indicative of the presence of *w* in reviews of *b*.

The collection of these values was referred to as a book's coordinates, as these values could be used to describe a book's position in a 30-dimensional space. Since each book occupied a coordinate in this book space, we used these coordinates as the basis of determining book similarity, by calculating the cosine similarity between two books. This similarity value was then used to cluster books by the weights of their feature tags. This process would later be used to determine the similarity between users, as well.

CHAPTER 4. RESULTS

4.1 K-Means Clustering

To generate book clusters using *k*-means clustering, *k* books were selected at random to be the centroids for the same number of initial clusters, with each cluster having the coordinates of its centroid book. In the clustering step, books were added to the clusters one at a time, by finding the book with the greatest similarity to an existing cluster and adding it to that cluster, until every book had been assigned to a cluster.

During the clustering process, the coordinates of each cluster center were considered to be the mean weight for each vector among all of its member books. The similarity between a book and a cluster was calculated by finding the cosine similarity between the book's coordinates, and that of the cluster's center. This cluster center was subsequently recalculated every time a book was added to the cluster.

As expected, the quality of the clusters using k-means was heavily dependent on the size of k, as well as the selection of initial centroids. In repeated runs with *k*=2, books tended to cluster by their sub-genres (fantasy and science fiction), except

in cases where both initial centroids were selected from the same sub-genre. Results of *k*-means clustering were more interesting as the value of k increased. At *k* values of 4 and above, book groups within each sub-genre clustered along nontrivial attributes began to emerge. Table 3 shows an example of the results generated by performing *k*-means clustering with *k*=5.

To identify the features present in each cluster, we compared the feature tag weight at the cluster center with its global weight. Clusters were considered to have a feature if the corresponding feature tag's weight within the cluster that was higher than its global weight. For each book cluster, we use boldface to represent features we consider to be their 'defining' features.

Table 3 Results of *k*-means clustering (*k*=5)

Cluster	Centroid	Books	Cluster Features
1	The Eyre Affair	The Eyre Affair, The Princess Bride, Going Postal, The Hitchhiker's Guide to the Galaxy, Small Gods	adventure, **entertaining**, **humor**, religion, small
2	The Stand	The Stand, Something Wicked This Way Comes, Homeland, Preludes and Nocturnes, Wicked, A Clockwork Orange, Watchmen, Animal Farm	**adult**, **dark**, **deep**, **evil**, political
3	A Game of Thrones	A Game of Thrones, The Eye of the World, Wizard's First Rule, Furies of Calderon Magician, Assassin's Apprentice, The Name of the Wind, Mistborn, The Way of Kings, Gardens of the Moon, The Chronicles of Thomas Covenant, the Unbeliever, Elric of Melniboné, The Fellowship of the Ring, The Belgariad, The Sword of Shannara Trilogy, The Crystal Cave, The Last Unicorn, A Spell for Chameleon, Stardust, Neverwhere, The Silmarillion, The Chronicles of Amber, Perdido Street Station, Dragonflight, The Mists of Avalon, Sunshine, The Once and Future King, Jonathan Strange and Mr. Norrell, American Gods, The Complete Chronicles of Conan, Kushiel's Dart, The Book of the New Sun, Watership Down, Outlander	adult, adventure, **battle**, compelling, complex, dark, **epic**, evil, **exciting**, **fantasy**, **hero**, intriguing, **magic**, simple, sex
4	World War Z	World War Z, The Moon is a Harsh Mistress, The Dispossessed, Starship Troopers, Red Mars, Foundation, Dune, The Martian Chronicles, Lucifer's Hammer, The Caves of Steel, I, Robot, The Time Machine, Anathem, Hyperion, The Forever War, Childhood's End, The Illustrated Man, Ringworld, The War of the Worlds, Rendezvous with Rama, Neuromancer, The Left Hand of Darkness, The Mote in God's Eye, Old Man's War, 2001, Shards of Honor, Consider Phlebas, Out of the Silent Planet, Contact, Do Androids Dream of Electric Sheep?, A Fire upon the Deep, Ender's Game, A Canticle for Leibowitz, Stranger in a Strange Land, Slaughterhouse Five, I Am Legend, Doomsday Book, The Diamond Age, Fahrenheit 451, 20,000 Leagues Under the Sea, Brave New World, Snow Crash, Journey to the Center of the Earth, Heir to the Empire, Frankenstein, The Handmaid's Tale, Cryptonomicon, Cat's Cradle, 1984, The Time Traveler's Wife, Flowers for Algernon	**alien**, **classic**, compelling, deep, entertaining, exciting, intriguing, **modern**, reality, religion, sad, **science**, social, **space**, **technology**, political
5	The Road	The Road, The Dark Tower	battle, **compelling**, **dark**, epic, reality, sad, simple

The clustering in Table 3 is made up of two large clusters (clusters 3 and 4), and three smaller ones (clusters 1, 2, and 5). A brief inspection of these results shows that for the most part, the features attributed to each cluster are accurate descriptors of books in the clusters.

The large clusters are the easiest to explain. Cluster 3, with A Game of Thrones, by George R. R. Martin as its centroid, is a large cluster composed of 'epic fantasy' books. All of the books in this cluster contain classic elements of epic fantasy, including a hero, magic, and adventure. The size of the cluster suggested that fantasy is a fairly formulaic genre, with many variations on the same themes of adventure, heroes, magic, and many of the other features associated with the cluster.

Something similar can be observed in cluster 4, which has features such as "alien", "science", "space", and "technology". However, this cluster is likely too large. Some of the books in the cluster: 20,000 Leagues Under the Sea, World War Z, and Frankenstein, to name a few, were set exclusively on Earth. While all of the books in the cluster explore hypotheticals rooted in reality (a post-apocalyptic zombie infestation, events in the *Star Wars* universe, dystopian versions of the near future, etc.), the books in this cluster could easily have been split into two or three smaller sub-clusters by a human familiar with the books.

The smaller clusters are more focused, with clear identifying attributes.

Cluster 1, with The Eyre Affair, by Jasper Fforde as its centroid contains books that are among the least 'serious' of the books in our data set: The Hitchhiker's Guide to the Galaxy is Douglas Adams's novel about a human who is plucked off of the planet Earth before it is demolished to make way for a galactic freeway. The Princess Bride is the story of a fairy tale gone wrong.

The second cluster is composed of books that obsess over themes of good and evil, with the possible exception of Wicked, which is lighter in tone than the others, being the tale of Wicked Witch of the West, from The Wizard of Oz, told from her point of view. Homeland, by R. A. Salvatore, was a particularly interesting fit in this cluster, since it can be considered to be fairly standard genre fantasy. The main character of Homeland is the member of a race of dark elves, who are evil by nature. Homeland is the story of his childhood in a society firmly rooted in evil, and his battle to retain his inner goodness, which is threatened by the evil of those around him. Homeland being clustered with The Stand and Animal Farm was extremely encouraging.

Stephen King's two novels, The Stand, and The Dark Tower are split between clusters 2 and 5, which is reasonable, since The Stand describes a post-apocalyptic struggle between good and evil, and The Dark Tower, while containing similar themes, does so over the course of a long journey. This description also fits The Road, the other member of the fifth cluster.

Although we were satisfied with the clusters generated by *k*-means, the clustering described above also reveals some of its deficiencies. Although *k*-means clustering is capable of producing satisfactory results, the possibility of generating poor clusters was not insignificant. Furthermore, having to specify the number of clusters to look for, in the form of *k*, has a clear impact on the quality of the results. Though there were ways to modify *k*-means clustering to take these concerns into consideration (additional clusters with human-selected centroids could be added, for example, increasing *k*), clusters generated by *k*-means clustering were unpredictable, and far too prone to generating poor clusters. Consequently, we began exploring hierarchical clustering as an alternative that would produce more consistent results, and help reduce the possibility of bad clustering.

4.2 Hierarchical Clustering

We began hierarchical clustering by setting each book as the centroid of its own cluster. Clusters were built up in successive rounds, by combining the two clusters with the greatest amount of similarity, resulting in increasingly fewer clusters, until all books became members of the same cluster. As in *k*-means clustering, the coordinates of the cluster centers were used, when determining the distance between two clusters. However, since there was no random selection of initial centroids, and no input parameter for the number of clusters to

generate, hierarchical clustering reliably produced the same results every time, eliminating the random elements of k-means clustering.

The hierarchical clustering process can be produced a clustering tree, showing the order in which clusters were combined. Figure 1 below shows an example of hierarchical clustering. Round 0 represents the preliminary round, with each book sole member its own cluster. Foundation and Dune are combined in round 1, forming cluster 6. Hyperion and Anathem are combined in round 2, forming cluster 7, and combined again in the next round with Lucifer's Hammer, forming cluster 8. Finally, clusters 6 and 8 are combined to form cluster 9 in step 4, ending the clustering process.

Since each round of clustering reduced the number of clusters by 1, we were able to observe the clustering results for any specified number of clusters, by viewing the clustering tree at round $t - n$, where t is the number of items being clustered, and *n* is the number of clusters desired. Referring again to Figure 1, we can see that at each round *r* of the clustering process, the value of $n + r$ is always equal to the value of *t*. We use *n* to refer to number of clusters in a hierarchical clustering, to differentiate these results from those produced by *k*-means clustering.

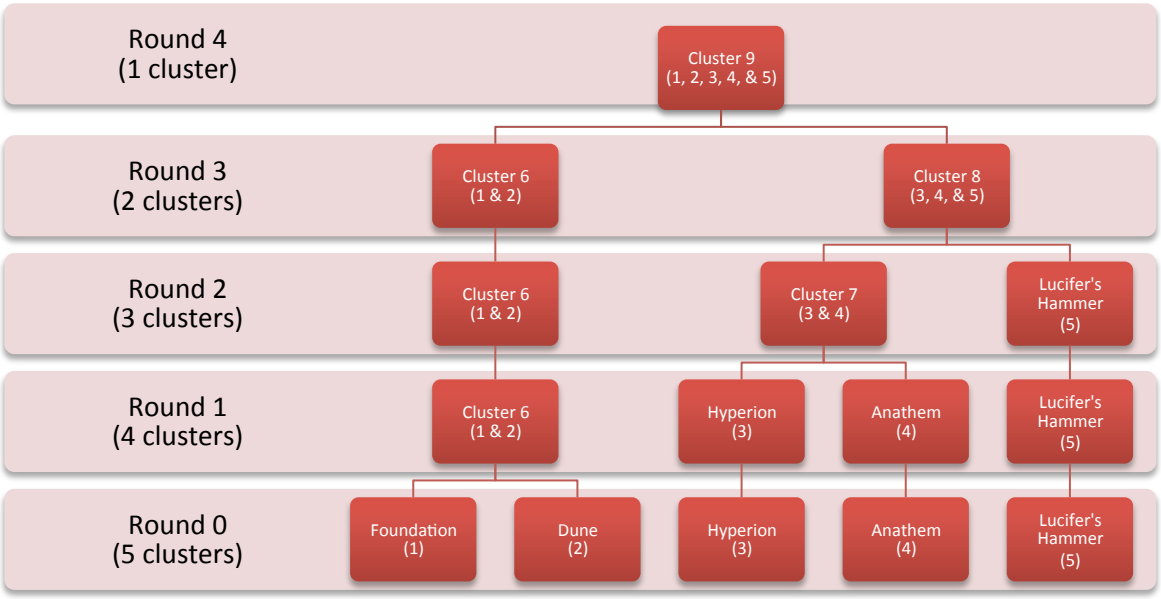

Figure 1 Results of sample hierarchical clustering run

Returning to the results of performing hierarchical clustering on the books on the NPR list, at $n = 2$, the clusters resembled the clusters produced by k-means when $k = 2$: books were split along their fantasy and science fiction sub-genres. At higher values of *n*, the clustering process was clearly not complete: the vast majority of clusters at these rounds only consist of one or two books. In these later stages of the clustering process, meaningful, non-trivial clusters were lost - a result of being merged into larger clusters. The clustering results for values of *n* between 5 and 10 were the best balance between incomplete clustering and over-clustering.

Table 4 shows our hierarchical clustering results when $n = 10$. As in Table 3, features in bold are those we consider to be cluster-defining features.

Table 4 Hierarchical clustering results (*n*=10)

Cluster	Books	Cluster Features
1	Small Gods	deep, **entertaining**, **humor**, **religion**, simple, small
2	The Time Traveler's Wife, Flowers for Algernon	**adult**, **sad**, simple, **sex**
3	Outlander, Kushiel's Dart	**adult**, **adventure**, compelling, complex, entertaining, epic, **exciting**, **fantasy**, hero, intriguing, religion, political, **sex**
4	The Dark Tower, The Road	compelling, **dark**, **epic**, reality, sad, simple
5	Watership Down, The Princess Bride	adult, **adventure**, **classic**, **entertaining**, **exciting**, humor
6	The Stand, Homeland, Something Wicked This Way Comes, Watchmen, Wicked, A Clockwork Orange	adult, battle, dark, deep, **evil**, hero, **social**, political
7	World War Z, Animal Farm, 1984, Fahrenheit 451	**classic**, modern, reality, sad, simple, **social**, **political**
8	Slaughterhouse Five, The Hitchhiker's Guide to the Galaxy, Going Postal, The Eyre Affair	**classic**, **entertaining**, **humor**, reality
9	Journey to the Centre of the Earth, 20,000 Leagues Under the Sea, I Am Legend, Frankenstein, Ender's Game, Heir to the Empire, Contact, The War of the Worlds, Ringworld, Childhood's End, The Mote in God's Eye, Rendezvous with Rama, The Forever War, Old Man's War, Out of the Silent Planet, 2001, The Illustrated Man, A Fire upon the Deep, Consider Phlebas, Shards of Honor, The Left Hand of Darkness, Starship Troopers, Red Mars, The Moon is a Harsh Mistress, The Dispossessed, The Time Machine, Neuromancer, The Martian Chronicles, I, Robot, The Caves of Steel, Lucifer's Hammer, Foundation, Dune, Hyperion, Anathem, Cat's Cradle, Do Androids Dream of Electric Sheep?, A Canticle for Leibowitz, The Handmaid's Tale, Stranger in a Strange Land, Brave New World, Doomsday Book, Cryptonomicon, Snow Crash, The Diamond Age	adventure, **alien**, **classic**, compelling, deep, entertaining, exciting, intriguing, **modern**, reality, religion, **science**, social, **space**, **technology**, political
10	The Book of the New Sun, The Complete Chronicles of Conan, American Gods, Preludes and, Nocturnes, Neverwhere, The Chronicles of Amber, Dragonflight, Perdido Street Station, A Spell for Chameleon, Jonathan Strange and Mr. Norrell, The Crystal Cave, The Last Unicorn, The Once and Future King, Stardust, Sunshine, The Mists of Avalon, The Sword of Shannara, Magician: Apprentice, Furies of Calderon, Assassin's Apprentice, The Name of the Wind, Mistborn: The Final Empire, A Game of Thrones, The Way of Kings, Gardens of the Moon, The Silmarillion, The Eye of the World, The Fellowship of the Ring, The Chronicles of Thomas Covenant, the Unbeliever, Elric of Melniboné, Wizard's First Rule, The Belgariad	adult, **adventure**, **battle**, compelling, complex, dark, **epic**, **evil**, **exciting**, **fantasy**, **hero**, humor, intriguing, **magic**, simple

As was the case with *k*-means clustering, most of the books that follow the general patterns of the science fiction and fantasy genres were lumped into two large clusters (clusters 9 and 10). Several of the books that were part of the smaller clusters were the same books in the smaller clusters in Table 3.

The presence of the 'classic' feature in the science fiction cluster (cluster 9) should also be discussed. As the third most heavily weighted feature tag with a weight of 0.134 (the feature tag with the next highest weight is 'dark', with a weight of 0.097), the classic feature tag is behind only the 'fantasy' and 'science' tags as the feature tag with the most influence over the clustering process. This is especially relevant to the science fiction genre, since the science fiction books on the NPR list are generally considered classic examples of science fiction genre fiction, written by authors such as Isaac Asimov, and Arthur C. Clarke. Recently written science fiction, such as The Time Traveler's Wife and The Dark Tower, are not yet considered to be classics (only time will tell). This undoubtedly reduced their similarity to the other science fiction books in the data set, reducing the likelihood of being merged into cluster 9.

By contrast, several recently written fantasy books are present in our data set. Several of the books in our data set were written in the past ten years: Kushiel's Dart was written in 2003, Outlander in 2005, and The Name of the Wind in 2007. The 'classic' feature is not attached to the large fantasy cluster, as was the case with the science fiction super-cluster. This suggests that fantasy is more

formulaic of a genre than science fiction, causing books in the genre to share enough features to reduce the impact of the powerful 'classic' feature tag in clustering.

Another anomalous result is worth discussing: Small Gods by Terry Pratchett remained a single-book cluster until the very last round of clustering. A blend of commentary on religion, satirical, humorous, and also fantasy, Small Gods is unique among books in the NPR list. Among the books in the data set, Cat's Cradle was most similar to Small Gods, with a cosine similarity of 0.728. As a basis for comparison, Journey to the Center of the Earth, and Twenty Leagues Under the Sea, which were combined in round 1 of clustering, were the two most similar books in our data set, with a cosine similarity of 0.968. The story of Small Gods's clustering misadventures can be explained by examining its neighbors.

Books with a cosine similarity above 0.5 (more similar than dissimilar) were considered to be neighbors. Small Gods has 10 neighbors, a low among the books on the NPR list. The average similarity between Small Gods and each of its neighbors was 0.575, second lowest among its peers. Since Small Gods was already a unique book among those in the dataset, it began with low cosine similarities to other books. Despite the existence of valid candidates for clustering (such as Cat's Cradle), as each of these neighbors were merged into other clusters, the attributes that made them similar to Small Gods were diluted in the process of merging, widening the gap between them.

Although the stability of hierarchical clustering was an obvious (and very important) advantage over *k*-means clusters, we felt it was important to ensure that books with viable clustering candidates were not passed over in the clustering process, as was the case with Small Gods.

4.3 Aggressive Hierarchical Clustering

In an effort to include books with few neighbors such as Small Gods in the clustering process, we proposed a far more aggressive clustering step in the initial stages of the clustering process. We did this by clustering each book with its closest neighbor in a pre-clustering round, allowing us to begin the actual process of hierarchical clustering with 50 clusters, each composed of a nearest-neighbor pair.

By performing this preliminary step, we hoped to make books with extreme weights in the clustering process more 'friendly' to its neighbors, while keeping each book's coordinates as close as possible to its original coordinates.
The results were promising, and an overall improvement over our first attempt at hierarchical clustering. Two books: Flowers for Algernon, and The Time Traveler's Wife. Though both books have slight tendencies towards science fiction, they are not widely considered to be science fiction novels (Flowers for Algernon is usually classified as classic literature, while The Time Traveler's Wife is usually described as contemporary literature, or even romance, before science

fiction). Given the unique classifications of these two books, we felt it was appropriate that they remained as a 2-book cluster until one of the last stages of clustering, where they were eventually combined with other books like Brave New World, 1984, and Fahrenheit 451: books with relatively similar faint elements of the science fiction and fantasy genres.

In addition to this preliminary step, we also made modifications to Bookmine to stop merging clusters when the cosine similarity between the closest clusters was less than a specified threshold, *t*. At *t*=0.5, books were clustered into two science fiction and fantasy sub-genres, as was the case with *k*-means clustering. At *t*=0.7, there were 9 book clusters: four clusters containing 2 books each, one cluster of books with social commentary, including books like Animal Farm and The Stand, one cluster of humorous classics, which included The Hitchhiker's Guide to the Galaxy and Slaughterhouse V, and one cluster of books about adventure, including Watership Down, The Princess Bride, and both books by Jules Verne. There were two large clusters. These were the clusters containing more prototypical science fiction and fantasy books.

After observing our clustering results at several threshold levels, we decided to use clustering results when *t*=0.75. At this level of clustering there were seven clusters with two books each, in a total of 13 book clusters, shown in Table 1.

Table 5 Aggressive hierarchical clustering using a threshold (*t*=0.75)

Cluster	Books	Cluster Features
1	The Time Traveler's Wife, Flowers for Algernon	**adult**, **sad**, simple, **sex**
2	Watership Down, The Princess Bride	adult, **adventure**, **classic**, **entertaining**, **exciting**, humor
3	The Dark Tower, The Road	battle, compelling, **dark**, **epic**, reality, sad, simple
4	Journey to the Center of the Earth, 20,000 Leagues Under the Sea	**adventure**, **classic**, deep, entertaining, exciting, modern, **science**, **technology**
5	Outlander, Kushiel's Dart	**adult**, **adventure**, compelling, complex, **entertaining**, epic, **exciting**, **fantasy**, hero, intriguing, religion, political, **sex**
6	The Complete Chronicles of Conan, Watchmen	adventure, **battle**, compelling, complex, dark, deep, entertaining, **evil**, fantasy, **hero**, modern, reality, simple, political
7	Small Gods, The Book of the New Sun	**epic**, **fantasy**, humor, reality, religion, simple, small, technology, sex
8	Doomsday Book, Cryptonomicon, Snow Crash, The Diamond Age	adventure, compelling, **complex**, entertaining, exciting, humor, **intriguing**, **modern**, **reality**, religion, **science**, **social**, **technology**, sex
9	The Mists of Avalon, American Gods, The Last Unicorn, The Once and Future King, Stardust, Sunshine, Preludes and Nocturnes, Neverwhere, Dragonflight, Perdido Street Station, The Way of Kings, Gardens of the Moon, Assassin's Apprentice, A Game of Thrones, The Name of the Wind, Mistborn, Magician: Apprentice, The Chronicles of Thomas Covenant, the Unbeliever, Elric of Melniboné, The Chronicles of Amber, The Silmarillion, Wizard's First Rule, The Belgariad, The Eye of the World, The Fellowship of the Ring, The Crystal Cave, Jonathan Strange and Mr. Norrell, The Sword of Shannara Trilogy, A Spell for Chameleon	adult, **adventure**, **battle**, compelling, complex, dark, **epic**, evil, **exciting**, **fantasy**, **hero**, humor, intriguing, **magic**, sad, small

Table 5 Continued.

Cluster	Books	Cluster Features
10	Homeland, Something Wicked This Way Comes, Wicked, A Clockwork Orange, Animal Farm, The Stand	**adult**, battle, dark, deep, **evil**, simple, **social**, **political**
11	I Am Legend, Frankenstein, Fahrenheit 451, 1984, The Handmaid's Tale, World War Z, Stranger in a Strange Land, Brave New World, The Moon is a Harsh Mistress, The Dispossessed, The Time Machine, The Left Hand of Darkness	**classic**, **modern**, **reality**, religion, sad, **science**, social, **political**, sex
12	Do Androids Dream of Electric Sheep?, A Canticle for Leibowitz, Cat's Cradle, Contact, Ender's Game, Heir to the Empire, Hyperion, Anathem, Foundation, Dune, Red Mars, Lucifer's Hammer, 2001, The Illustrated Man, I, Robot, The Caves of Steel, Neuromancer, The Martian Chronicles, Starship Troopers, The Forever War, Ringworld, Old Man's War, The Mote in God's Eye, Rendezvous with Rama, The War of the Worlds, Childhood's End, Consider Phlebas, Shards of Honor, A Fire upon the Deep, Out of the Silent Planet	**alien**, battle, **classic**, compelling, complex, deep, entertaining, exciting, intriguing, reality, religion, **science**, small, social, **space**, **technology**, political
13	Slaughterhouse Five, or the Children's Crusade, The Hitchhiker's Guide to the Galaxy, Going Postal, The Eyre Affair	**classic**, **entertaining**, **humor**, reality

Several of the clusters from before are still present in Table 5. Although the science fiction cluster super-cluster from earlier has been broken into two smaller (but still relatively large) clusters, the fantasy books in the data set remain in one massive cluster: cluster 9 contains 30 books, nearly a third of the books in our data set.

Although seven of the clusters are still only made up of two books each, it was difficult to combine these smaller clusters together among themselves, and we believed the similarity threshold had kept most of these books separate from the larger clusters for a reason, as in the case of cluster 1. Since this thesis is concerned with assessing the validity of this method of clustering books with

minimal human interaction, we were satisfied enough with the clustering results to move forward with the next step of our study. If nothing else, these smaller clusters highlighted the limitations of working with a smaller data set. In our future work section, we discuss steps to address this.

4.4 Cluster Evaluation

Thus far, we have evaluated the results of book clustering using our knowledge of the books in the data set. To allow us to evaluate our clustering method objectively, we needed to make user of our user rating data. Since this particular domain is so subjective by nature, we felt it was appropriate to evaluate the results of book clustering by looking for affirmation of their accuracy in our user data.

Other studies [13][14] have used humans to evaluate the quality of feature extraction, often by providing feedback about the results, often through interviews or surveys. However, this process can be time consuming, placing an effective limit on the amount of data that can be collected as feedback. Also, feedback gained from this type of evaluation is still ultimately the result of subjective opinion. While this is an inevitable consequence of evaluating any solution in an opinion-based domain, the effects of subjectivity can be 'smoothed' to some extent, by increasing the size of *n*. Put simply, one person may hate a particular book recommendation, but as more people give feedback on the same

recommendation, the percentage of people who respond favorably to the recommendation approaches the 'true' accuracy of the recommendation.

Of course, relative to each person we make predictions about, each prediction is subjective. Asking multiple people to give feedback about the 'same' user-specific prediction adds little value, since the user in question is the only person who can truly assess accuracy of each prediction. Here again, we are able to increase the size of *n* to cope: by making more predictions, and evaluating their accuracy as a group, we are able to make judgments about the accuracy of the approach, based on generalizations aggregated from all predictions we have made, using this approach. The amount of user data at our disposal is what allows us to perform this type of evaluation.

Previously, in our data collection process, we collected every review that had been written about the books on the NPR 100 list, including the Goodreads user ID of the author of each review. The number of reviews written by each of the 58,493 users in our data set, who were given new ID numbers.

All reviews in our data set were sorted by user, which allowed us to mine each user in the same way we mined books, looking for weights of the same feature tags used for book clustering. Mining user reviews with the same set of features was a natural extension of our work in clustering books. Our successful extraction of meaningful content features about books by mining review text, lent

credibility to our baseline assumption: that reviews users write about books they read contain descriptions of the book's content. Similarly, we believed that by mining the text of a user's reviews and looking for those same features, we could make reasonable predictions about the type of book a particular user tends to read. By performing the same feature identification for a user, and looking for a correlation between books they have read, and books that Bookmine thinks they are likely to read, we would be able to evaluate the performance of our program.

The number of reviews written by each user was much smaller than the number of reviews written about any of the books in our data set. Additionally, the quality of each user's reviews varied wildly. Some users wrote reviews that were several hundred words in length for each of the books they read, while others wrote only a few words each. The quality of reviews written by the same user varied from book to book, as well. Reviews written for books that received one or five star ratings tended to be significantly longer than books receiving three or four star ratings.

It became apparent early in the process that some sort of lower bound on the number of reviews a user had written would need to be established, before mining the user's reviews. For the vast majority of our users, there simply was not enough review text to extract any meaningful information. Therefore, we decided to only mine the reviews of users who had reviewed at least 20 of the books on the NPR 100 list. In doing this, we were trying to minimize the number

of users who would be far less likely to have any feature tags used in their reviews. For users with fewer reviews (particularly those with five or less reviews, which were the vast majority of our users), a review containing a single feature tag word would carry an inordinate amount of weight, to the point where the user would be end up being defined by the one or two feature tags they had used in their reviews. Furthermore, the tendency of these users to be one-dimensional could wreak havoc in the clustering step, by being one-dimensional to the point of being unable to be clustered (like Small Gods was in hierarchical clustering), or by skewing the center of any cluster they joined.

After removing these users, our user pool shrank from 58,493 to 182 users, who had written a total of 4,715 reviews, about 25 each. The fact that a particular user can have so much to say about one book, while being reticent about another was a feature of user reviews that we felt would increase the effectiveness of identifying the features that users were interested in, by mining their book reviews. Since more text is written about books that evoke strong feelings in a user, features that the user is especially interested (or possibly disinterested) in are likely to have a higher occurrence in the user's reviews. In addition, users are far more likely to express a positive opinion than a negative one: of the 1,583 reviews that received an extreme rating on either end of the rating spectrum (one or five stars), only 10% of these reviews were one star reviews. Users were, in fact, nine times as likely to express an extreme positive opinion.

The review data of the remaining 182 users was mined, using the same process previously used to mine book reviews. As with books, this step allowed us to describe users using a set of 30 weight vectors. Due to the concerns we described regarding entities with a high proportion of feature tags with zero weight, we excluded users who had not used at least three of the words associated with feature tags in their reviews. This further reduced the number of users to 168, and the number of reviews to 4,396.

At this point, we had a set of coordinates to associate with each of the remaining 162 users, whose reviews we considered to contain a sufficient level of information. These user coordinates allowed us to begin to determine the quality of the feature mining results obtained previously. We proposed to do this by looking for a global correlation between a user's cosine similarity to a book, and the rating the book received.

The first step of this evaluation process was to confirm our assertion that users were more likely to write about the book features that they like. This was verified by using each user's coordinates to find the books that had been identified as having those characteristics, by calculating the cosine similarity between the user's coordinates, and those of each book in the data set. Next, we examined the ratings the user gave to those books. Specifically, we were looking for the ratings users gave books to trend upwards as the similarity between the two increased. Again, the subjectivity user opinion makes it difficult to draw any

conclusions by examining the ratings of any one user, but when examined as a whole, a more generalized analysis allowed us to notice a pattern.

We used a threshold value to filter the ratings, allowing us to examine user ratings of books with a certain similarity. Table 6 shows the number of ratings at each threshold level. It can be seen that there are low numbers of ratings at the extremes (no ratings at all that satisfy $s = 0.95$ and above, while there is only one rating with a similarity of 0), while the majority of ratings in our data set were distributed at threshold values around $s = 0.5$.

Table 6 Ratings at each similarity threshold *s*

s	**# of ratings satisfying *s***	
	Not cumulative	Cumulative
1.00	0	0
0.95	0	0
0.90	1	1
0.85	36	37
0.80	84	121
0.75	195	316
0.70	343	659
0.65	412	1071
0.60	466	1537
0.55	485	2022
0.50	479	2501
0.45	442	2943
0.40	432	3375
0.35	341	3716
0.30	243	3959
0.25	177	4136
0.20	132	4268
0.15	80	4348
0.10	36	4384
0.05	11	4395
0.00	1	4396

At each threshold value *s*, we looked for all pairs of users and books that had this level of similarity.

For example, to find the average rating at $s = 0.6$, we calculated the cosine similarity between each user, and all 100 books in the data set. For each of these books, if the cosine similarity between the book's coordinates and the user's coordinates are 0.6 or above (and if the user had given this book a rating), we added the rating associated with this user-book pair to a list. After finding all of these pairs that exist in the data set, the ratings associated with each of these pairs were averaged, resulting in value that represented the average rating given by users across the entire data set to books with a cosine similarity to them of *s* or higher. Then, *s* was reduced by 0.05, and the process was repeated for each new value of *s*.

Table 7 shows the results of this process, which reveals a clear correlation between similarity and rating. The ratings dip at $s = 0.85$ is likely due to the low number of ratings that meet this level of threshold (this can be seen in Table 6). At $s = 0.9$, there was only a single rating, although it was a four-star rating. There are only 37 ratings at $s = 0.85$, inclusive of the rating at $s = 0.9$.

Table 7 Cumulative average rating by similarity

Similarity *(s)*	**Average rating**
1.00	N/A
0.95	N/A
0.90	4.0000
0.85	3.8108
0.80	3.9835
0.75	3.8196
0.70	3.7102
0.65	3.6760
0.60	3.6285
0.55	3.6390
0.50	3.6329
0.45	3.6449
0.40	3.6388
0.35	3.6550
0.30	3.6585
0.25	3.6639
0.20	3.6664
0.15	3.6667
0.10	3.6688
0.05	3.6696
0.00	3.6697

Average rating also increases in the lower values of *s*. However, this is less significant than the ratings at higher values of *s*, since we are only concerned with the correlation between rating and similarity. This is not really an issue, since there are very few ratings that are this dissimilar from the users who wrote them. Also, similarity has far less significance at these lower values of *s*: the difference between a similarity value of 0.85 and 0.75 is far more meaningful than a difference between 0.15 and 0.05. It should also be noted that the average rating at $s = 0$ is the equivalent of the average of all ratings given by users whose reviews were mined. Plotting the data in Table 7 produced a curve showing this correlation, shown in Figure 2.

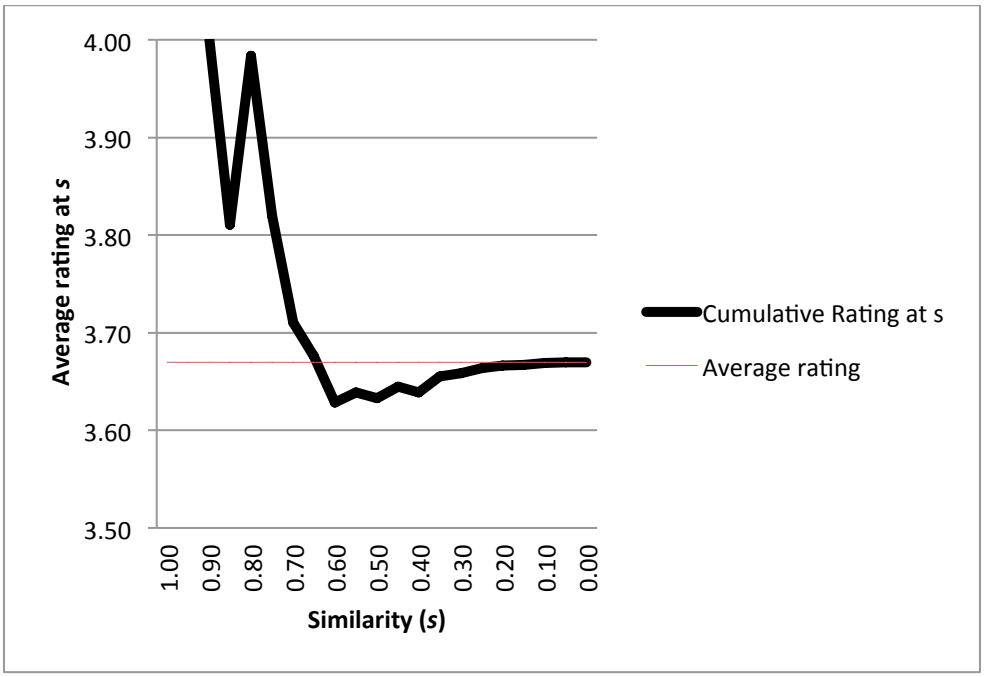

Figure 2 Plotted correlation between similarity and rating

Though Figure 2 clearly shows that users give higher ratings to books they were similar to, we were curious to what extent this was true. Since each user has different rating habits, reflected by the average of all ratings they gave, a four-star rating did not necessarily have the same significance for two different users. In order to evaluate the extent that these higher ratings translated into an accurate measure of user opinion, we decided to repeat the process, using a simple over/under test for each review collected. To make this comparison, each user's reviews are compared to the average of all of that user's ratings. At each threshold, we determined the percentage of ratings given by users that was at least as high as the user's average review.

For each threshold s, the percentage of books rated higher than the user's average rating can be thought of as a 'hit' rate, similar to the metrics used in recommendation systems to evaluate whether a user responded favorably to a set of recommendations. We referred to this metric as the net positive at some threshold s. The net positivity also acts as a measure of confidence: for books that have a similarity to a user of 0.75, we are 61.7% sure that they will like the book more than other books they have read. However, in calculating this metric, we discovered that the discrete nature of the Goodreads rating system was somewhat of a problem. Since ratings on Goodreads can only be an integer between one and five, there is no way for a user to give a book a rating between any of the discrete values. If a user who feels a book is should be rated somewhere between three and four stars, they must make a choice between the two.

This had the effect of negatively skewing the positivity measure for many users. For example, if a user rated a total of ten books, with 9 receiving a four-star rating, and one book receiving a five star rating, that user's average review score would be 4.1, causing all of that user's four-star reviews to be flagged as negative. We felt this caused too many of these borderline cases to be counted as instances where the user did not like the book. To compensate, we rounded a user's average rating down to the nearest integer, but only for users with average ratings less than half a star above this number (an average rating of 3.2 would have this rounded down to 3, but an average rating of 3.78 would be unchanged).

Table 8 shows the net positivity observed at each value of *s*. For the purposes of calculating net positivity, books a user had indicated they were interested in reading (and therefore had not yet rated) were assumed to count as positive, relative to their average rating. Although a case can be made for the omission of these data points, the goal of this analysis was to determine the likelihood of a user being interested in a book, based on similarity. We felt the act of indicating interest in a book was enough to count as an instance of a positive rating, considering each of the users in our data set had expressed opinions about at least 20 of the books in our data set.

Since net positivity is our measure of the accuracy of this method of feature identification, plotting the relationship between net positivity at each similarity threshold shows a general degradation in net positivity as similarity decreases (this can be seen in Figure 3). As in Table 7, the single four-star rating at $s = 0.9$ skews the data at this threshold, producing an overly optimistic 100% net positivity.

Table 8 Net positivity by similarity

Similarity	Net Positivity at *s*
1.00	N/A
0.95	N/A
0.90	1.0000
0.85	0.5676
0.80	0.6033
0.75	0.5633
0.70	0.5114
0.65	0.5163
0.60	0.5166
0.55	0.5208
0.50	0.5202
0.45	0.5229
0.40	0.5200
0.35	0.5231
0.30	0.5256
0.25	0.5285
0.20	0.5298
0.15	0.5315
0.10	0.5324
0.05	0.5329
0.00	0.5328

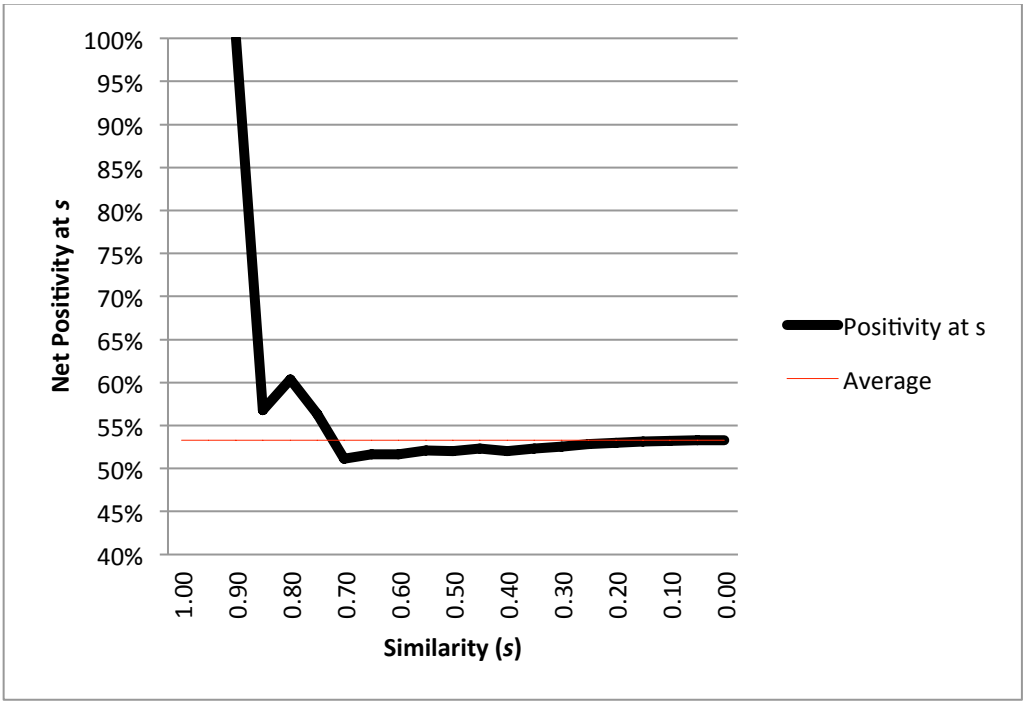

Figure 3 Net positivity by similarity

At lower values of *s*, a small increase in net positivity can be observed. This is due to the cumulative nature of this calculation. Therefore, we can deduce that there are sufficient amounts of positive ratings at lower values of *s* to pull the net positivity back to the global average. This can be confirmed by examining positivity at each value of *s*, in a noncumulative manner, shown in Figure 4.

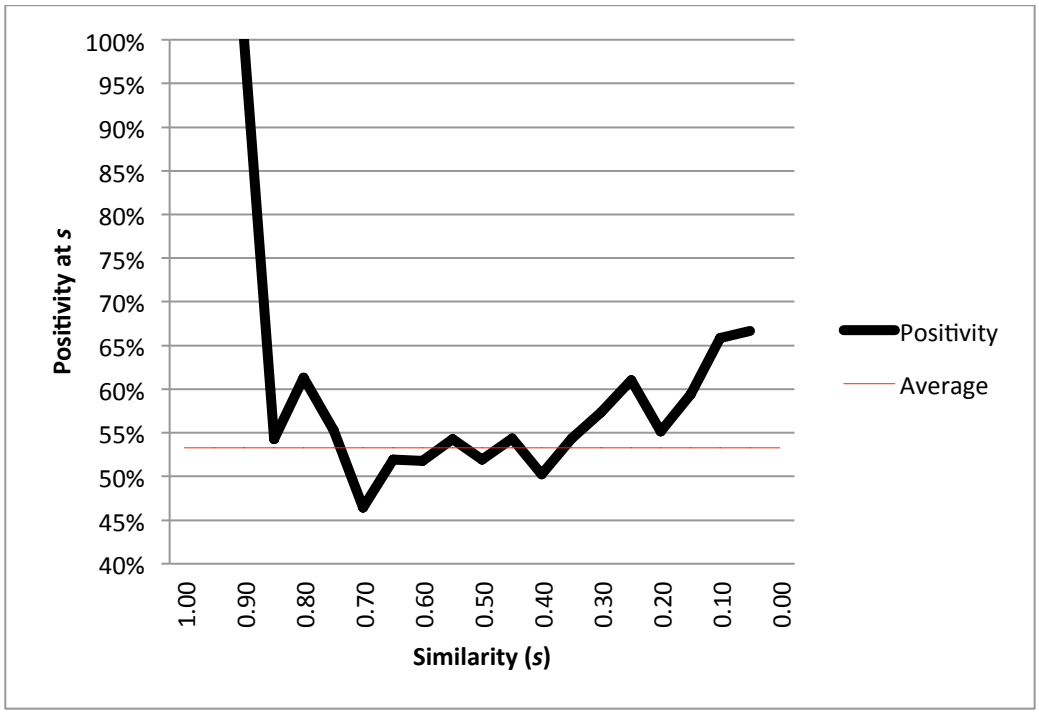

Figure 4 Non-cumulative positivity by *s*

We were interested to see if this pattern would also be present if we were to perform the same analysis using book clusters. We were aware that making comparisons to book clusters would likely degrade the quality of our results, due to the weight values of the book cluster being derived from the mean of all its

member books. This 'centering' of cluster coordinates caused books that are most similar to the user to have less similarity, while bringing books that are the least similar to have a higher similarity. For example, given any two books b_1 and b_2, which are similar enough to each other to be candidates for clustering in the clustering step. Between these two books, we can assume that one is more similar to user *u* than the other (let us assume that b_1 is more similar to *u*). It is theoretically possible that they are equally similar to *u*, but this is extremely unlikely (this is actually the ideal case). Once clustered into cluster *c*, when calculating the similarity between *u* and *c*, b_1 will have become less similar to *u*, while b_2 will have become more similar. This is the tradeoff inherent in performing any type of grouping: the exchange of accuracy for the ability to make broader generalizations.

Given a sufficient level of clustering, results were almost certain to degrade, due to the centering problem described previously. The goal, then, was to find the level of clustering in which the results became unreliable: to find the point where the process broke down. Table 9 shows the result of our analysis of net positivity at various levels of book clustering. The results from previously, with no book clustering at all, are included for comparison.

Table 9 Net positivity at various levels of book clustering

s	No clustering	50 clusters	40 clusters	30 clusters	25 clusters	20 clusters	15 clusters	13 clusters
1.00	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
0.95	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
0.90	1.0000	1.0000	0.5000	0.5000	0.5882	0.6154	0.0000	0.0000
0.85	0.5676	0.6000	0.5179	0.4464	0.5818	0.5692	0.5588	0.4444
0.80	0.6033	0.5226	0.5309	0.5202	0.4819	0.4850	0.4629	0.4631
0.75	0.5633	0.5330	0.5343	0.5073	0.5176	0.4972	0.5151	0.5154
0.70	0.5114	0.5115	0.5197	0.5290	0.5286	0.5256	0.5377	0.5371
0.65	0.5163	0.5201	0.5176	0.5172	0.5169	0.5099	0.5116	0.5111
0.60	0.5166	0.5239	0.5192	0.5137	0.5120	0.5139	0.5143	0.5157
0.55	0.5208	0.5277	0.5287	0.5282	0.5252	0.5267	0.5277	0.5282
0.50	0.5202	0.5263	0.5236	0.5261	0.5261	0.5265	0.5277	0.5288
0.45	0.5229	0.5263	0.5260	0.5255	0.5253	0.5230	0.5244	0.5254
0.40	0.5200	0.5221	0.5218	0.5223	0.5225	0.5243	0.5243	0.5239
0.35	0.5231	0.5257	0.5260	0.5264	0.5258	0.5261	0.5269	0.5270
0.30	0.5256	0.5281	0.5287	0.5291	0.5292	0.5280	0.5287	0.5289
0.25	0.5285	0.5293	0.5295	0.5288	0.5299	0.5300	0.5316	0.5320
0.20	0.5298	0.5305	0.5310	0.5321	0.5318	0.5318	0.5320	0.5323
0.15	0.5315	0.5320	0.5319	0.5321	0.5325	0.5325	0.5326	0.5327
0.10	0.5324	0.5326	0.5324	0.5325	0.5325	0.5325	0.5325	0.5325
0.05	0.5329	0.5328	0.5328	0.5328	0.5328	0.5328	0.5328	0.5328
0.00	0.5328	0.5328	0.5328	0.5328	0.5328	0.5328	0.5328	0.5328

There is a clear trend in the results, showing lower levels of net positivity as clustering (and cluster sizes) increases. We also noticed that for the most part, there was little variance among net positivity values. In fact, given a sufficiently large number of ratings that meet a particular similarity threshold, there is very little deviation from the global net positivity of 0.5328. This can be observed at $s = 0.5$, while net positivity is much more volatile at higher values of *s*, where there are far fewer ratings. In order to better visualize these results, we used a heat map visualize the table shown in Table 9, shown in Figure 5.

s	100	50	45	40	35	30	25	20	15	13
1.00										
0.95										
0.90	1.0000	1.0000	0.5000	0.5000	0.5000	0.5000	0.5882	0.6154	0.0000	0.0000
0.85	0.5676	0.6000	0.6078	0.5179	0.4906	0.4464	0.5818	0.5692	0.5588	0.4444
0.80	0.6033	0.5226	0.5244	0.5309	0.5407	0.5202	0.4819	0.4850	0.4629	0.4631
0.75	0.5633	0.5330	0.5262	0.5343	0.5327	0.5073	0.5176	0.4972	0.5151	0.5154
0.70	0.5114	0.5115	0.5145	0.5197	0.5218	0.5290	0.5286	0.5256	0.5377	0.5371
0.65	0.5163	0.5201	0.5165	0.5176	0.5152	0.5172	0.5169	0.5099	0.5116	0.5111
0.60	0.5166	0.5239	0.5213	0.5192	0.5181	0.5137	0.5120	0.5139	0.5143	0.5157
0.55	0.5208	0.5277	0.5279	0.5287	0.5297	0.5282	0.5252	0.5267	0.5277	0.5282
0.50	0.5202	0.5263	0.5261	0.5236	0.5260	0.5261	0.5261	0.5265	0.5277	0.5288
0.45	0.5229	0.5263	0.5267	0.5260	0.5267	0.5255	0.5253	0.5230	0.5244	0.5254
0.40	0.5200	0.5221	0.5213	0.5218	0.5235	0.5223	0.5225	0.5243	0.5243	0.5239
0.35	0.5231	0.5257	0.5260	0.5260	0.5258	0.5264	0.5258	0.5261	0.5269	0.5270
0.30	0.5256	0.5281	0.5287	0.5287	0.5283	0.5291	0.5292	0.5280	0.5287	0.5289
0.25	0.5285	0.5293	0.5293	0.5295	0.5283	0.5288	0.5299	0.5300	0.5316	0.5320
0.20	0.5298	0.5305	0.5309	0.5310	0.5311	0.5321	0.5318	0.5318	0.5320	0.5323
0.15	0.5315	0.5320	0.5319	0.5319	0.5320	0.5321	0.5325	0.5325	0.5326	0.5327
0.10	0.5324	0.5326	0.5324	0.5324	0.5327	0.5325	0.5325	0.5325	0.5325	0.5325
0.05	0.5329	0.5328	0.5328	0.5328	0.5328	0.5328	0.5328	0.5328	0.5328	0.5328
0.00	0.5328	0.5328	0.5328	0.5328	0.5328	0.5328	0.5328	0.5328	0.5328	0.5328

Figure 5 Heat map showing net positivity at various levels of book clustering

As can be seen in the heat map, results are most meaningful at $s = 0.75$ and above. Additionally, there is relatively little variance from the global net positivity at values of *s* with a large amount of ratings. In contrast, there are low numbers of ratings at $s = 0.85$ and $s = 0.9$, causing the net positivity at these thresholds varies greatly at different levels of book clustering.

We repeated the analysis with our user clusters, formed using the same aggressive hierarchical clustering process we used for book clustering. Each user was clustered with its nearest neighbor in a preliminary clustering step,

followed by a series of cluster merges, with each round reducing the number of clusters by one.

Unlike with book clustering, we were unable to evaluate the quality of user clusters by examining the features of their reviews, as nontrivial attributes for users could not be evaluated by comparing these attributes to domain knowledge, since no such knowledge exists. Therefore, we were unable to determine the effectiveness of our user clustering by examining them directly. Instead, we reused the measure of net positivity to determine the quality of at the net positivity at various degrees of clustering. As with book clusters, we expected to see a decline in net positivity as the amount of clustering increased.

Figure 6 shows the net positivity at various levels of user clustering. As was the case with book clustering, results are less stable at higher values of *s*, though it is interesting to see that results seem more resilient to clustering than book clustering.

s	168	84	75	60	45	30
1.00	N/A	N/A	N/A	N/A	N/A	N/A
0.95	N/A	N/A	N/A	N/A	N/A	N/A
0.90	1.0000	0.6000	0.6154	0.5185	0.5366	0.4146
0.85	0.5676	0.6275	0.5976	0.6143	0.5721	0.4762
0.80	0.6033	0.5902	0.5981	0.5645	0.5482	0.5688
0.75	0.5633	0.5251	0.5285	0.5266	0.5200	0.5481
0.70	0.5114	0.5269	0.5230	0.5251	0.5317	0.5416
0.65	0.5163	0.5075	0.5132	0.5163	0.5191	0.5193
0.60	0.5166	0.5148	0.5167	0.5177	0.5196	0.5149
0.55	0.5208	0.5188	0.5185	0.5179	0.5173	0.5184
0.50	0.5202	0.5223	0.5227	0.5206	0.5198	0.5222
0.45	0.5229	0.5271	0.5273	0.5248	0.5254	0.5261
0.40	0.5200	0.5276	0.5277	0.5279	0.5265	0.5285
0.35	0.5231	0.5300	0.5300	0.5300	0.5296	0.5304
0.30	0.5256	0.5304	0.5310	0.5321	0.5320	0.5312
0.25	0.5285	0.5314	0.5313	0.5317	0.5325	0.5317
0.20	0.5298	0.5329	0.5327	0.5329	0.5332	0.5327
0.15	0.5315	0.5323	0.5326	0.5326	0.5326	0.5326
0.10	0.5324	0.5328	0.5328	0.5328	0.5328	0.5328
0.05	0.5329	0.5328	0.5328	0.5328	0.5328	0.5328
0.00	0.5328	0.5328	0.5328	0.5328	0.5328	0.5328

Figure 6 Heat map showing net positivity at various levels of user clustering

CHAPTER 5. CONCLUSION

In this thesis, we proposed a method to mine attributes from book reviews by identifying book features in the review text. Our mining process produced a set of vectored coordinates for each book and user in our data set, with values corresponding to the global term frequency-inverse document frequency of each of the feature tag words selected to describe the books in our data set. Additionally, we demonstrated that the features identified through this process correspond to domain knowledge.

We were able to use these book coordinates to achieve meaningful clustering of the books in our data set according to these nontrivial features. Due to our hierarchical method of clustering, and we were able to observe clustering results at various degrees of clustering. Furthermore, we have demonstrated that there is a correlation between the features mined from a particular book's reviews, and the features expressed in a user's reviews. This was observed by comparing the ratings given to books with a high similarity to the user who rated them, with ratings of books that have a low similarity to the user who gave the rating, in a measure we termed 'net positivity'.

Since users and books had comparable coordinates, we were able to make determine the similarity between any combinations of these two types of entities. By determining the net positivity at various levels of book and user clustering, we observed a degradation of net positivity as clustering increased. It became clear that books had a limited tolerance of clustering before net positivity became uncorrelated with similarity.

In the future, we plan work to study the limitations of our methods. In particular, we would like to develop a solution to clustering users that is not quite so reliant on having large amounts of review data written by the same user. A hybrid approach involving user ratings, and features mined from books could allow for improved accuracy in user clustering. We are also interested in exploring other methods of measuring similarity between books, including clustering that can be performed at-will, taking a user's preferences into account. Finally, we would like to study possible applications of this method of book and user clustering. This includes making predictions about user preferences, in the form of recommendations.

LIST OF REFERENCES

LIST OF REFERENCES

[1] Bowker [Internet], Available online: http://www.bowkerinfo.com/bowker/IndustryStats2010.pdf 04/23/2012

[2] Goodreads is an online community of readers [Internet]. Available at: www.goodreads.com

[3] NPR Top 100 Science Fiction, Fantasy books [Internet]. Available at: http://www.npr.org/2011/08/11/139085843/your-picks-top-100-science-fiction-fantasy-books

[4] G. Salton and C. Buckley. “Term-weighting approaches in automatic text retrieval,” in *Information Processing and Management*, vol. 24, no. 5, pp. 513-523, 1988.

[5] G. Salton. Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer. Reading, Addison-Wesley, 1989.

[6] R. Feldman and I. Dagan. “Knowledge discovery in textual databases (KDT),” in *Proceedings of the First International Conference on Knowledge Discovery and Data Mining*, 1995, pp. 112-117.

[7] R. Feldman, I. Dagan, and H. Hirsh, “Mining Text Using Keyword Distributions,” in *Journal of Intelligent Information Systems: Integrating Artificial Intelligence and Database Technologies*, pp. 291-300, 1998.

[8] C. D. Fellbaum. *WordNet: An Electronic Lexical Database*. Cambridge, MA: MIT Press, 1998.

[9] T. Hofmann. “Probabilistic Latent Semantic Indexing,” in *Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence*, pp. 289-296, 1999.

[10] D. Oelke, P. Bak, D. Keim, M. Last, and G. Danon. “Visual evaluation of text features for document summarization and analysis,” in *IEEE Symposium on Visual Analytics and Technology*, pp. 75-82, 2008.

[11] M. L. Gregory, N. Chinchor, P. Whitney, R. Carter, E. Hetzler, and A. Turner. "User-directed sentiment analysis: visualizing the affective content of documents," in *Proceedings of the Workshop on Sentiment and Subjectivity in Text*, ser. SST '06, pp. 23-30, 2006.

[12] Q. You, S. Fang, and P. Ebright. "Iterative visual clustering for Unstructured Text Mining," in *International Symposium on Biocomputing*, Calicut, Kerala, India, 2010.

[13] B. Pang and L. Lee, "Opinion Mining and Sentiment Analysis," in *Foundations and Trends in Information Retrieval*, vol. 2, no. 1-2, pp. 1-135, 2008.

[14] B. Pang and L. Lee. "Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales," in *Proceedings of Computational Linguistics*, pp. 115-124, 2005.

[15] S. Sahar. "Interestingness via what is not interesting," in *The Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 332-336, 1999.

[16] A. Silberschatz and A. Tuzhilin. "What makes patterns interesting in knowledge discovery systems," in *IEEE Transactions on Knowledge and Data Engineering*, vol. 8, issue 6, pp. 970-974, 1996.

[17] J. Blitzer, M. Dredze, and F. Pereira. "Biographies, Bollywood, Boom-Boxes, and Blenders: Domain adaptation for sentiment classification," in *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics*, pp. 440-447, 2007.

[18] F. Wanner, J. Fuchs, D. Oelke, and D. Keim. "Are my children old enough to read these books? Age suitability Analysis," in *Polibits*, vol. 43, pp. 93-100, 2011.

[19] G. Qian, S. Sural, Y. Gu, and S. Pramanik. "Similarity between Euclidean and cosine angle distance for nearest neighbor queries," in *Proceedings of the 2004 ACM symposium on Applied computing*, pp. 1232-1237, 2004.

[20] R. Mihalcea, C. Corley, and C. Strapparava. "Corpus-based and Knowledge-based Measures of Text Semantic Similarity," in *Proceedings of the 21st National Conference on Artificial Intelligence*. AAAI Press, 2006.

[21] D. Goldberg, D. Nichols, B. M. Oki, and D. Terry. "Using Collaborative Filtering to Weave an Information Tapestry," in *Communications of the ACM*, vol. 35, no. 12, pp. 51-60, 1992.

[22] W. Hill, L. Stead, M. Rosenstein, and G. Furnas, “Recommending and evaluating choices in virtual community of use,” in *Proceedings of CHI'95*, pp. 194-201, 1995.

[23] B. Liu, W. Hsu, L.-F. Mun, and H. Lee. “Finding interesting patterns using user expectations,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 11, no. 6, pp. 817-832, 1999.

[24] D. Gillick and Y. Liu. “Non-expert evaluation of summarization systems is risky,” in *NAACL Workshop on Creating Speech and Language Data with Amazon’s Mechanical Turk*, pp. 148-151, 2010.

[25] L. H. Ungar, and D. P. Foster. “Clustering methods for collaborative filtering” in *Recommender Systems*. *Papers from 1998 Workshop. Technical Report WS-98-08*. AAAI Press, 1998.

[26] R. J. Mooney and L. Roy. “Content-based Book Recommending Using Learning for Text,” in *Proceedings of the Fifth ACM Conference on Digital Libraries*, pp. 194-204, 2000.

[27] G. Adomavicius and A. Tuzhilin. “Multidimensional recommender systems: a data warehousing approach,” in *Proceedings of the 2nd International Workshop on Electronic Commerce (WELCOM’01). Lecture Notes in Computer Science*, vol. 2232, Springer, 2001b.

[28] J. Alspector, A. Kolcz, and N. Karunanithi. “Comparing feature-based and clique-based user models for movie selection,” in *Proceedings of the Third ACM Conference on Digital Libraries,* pp. 11-18, Pittsburgh, PA, June 1998.

[29] L. Campos, J. Fernández-Luna, J. Huete, M. Rueda-Morales, “Combining content-based and collaborative recommendations: A hybrid approach based on Bayesian networks,” in *International Journal of Approximate Reasoning*, vol. 51, no. 7, 2010.

[30] J. Han, M. Kamber, and J. Pei. *Data Mining: Concepts and Techniques.* San Francisco, CA: Morgan Kauffmann Publishers, 2011.