PickyMuch: An Analytic Visualization of Book Recommendations
Mary Elaine Ramos Malit
Toronto, Canada
elainemalit@cs.toronto.edu

ABSTRACT
Many book recommendation systems employ a minimalist approach when they present their results to users, often using just a plain list. While a simplistic design reduces cognitive load on the viewer, it also fails to provide a mechanism for users to compare recommended items. This lack of functionality forces the user to search for more information about each item and manually compare their features (ie. average rating, length, genres). This process usually involves navigating to the page of individual items, which can be tedious especially if the list is long. I introduce PickyMuch, a web-based analytic visualization prototype for book recommendations. PickyMuch provides several features that facilitate item comparison, such as sorting, filtering, plotting and item-tracking across pages. With PickyMuch, I explore the research question of whether a richer visualization of recommended items can help people decide more efficiently what to read next.

ACM Classification Keywords
H.5.2. Information Interfaces and Presentation (e.g. HCI): User interfaces

Author Keywords
User interfaces for recommendation results; visualizing document collections; web development

INTRODUCTION
Discovering recommendations for books is now easier than ever. One only has to go to an online vendor like Amazon or a social network like Goodreads to find hundreds, if not thousands, of books to choose from. Hence it is quite surprising that recommendation sites do not provide a richer platform for users to explore and compare these recommendations. Some sites may offer search engines that support dynamic queries, or menus with categories like “Mystery” or “Romance” to help users find books. However, recommendations based on specific books in particular, like Amazon’s “Customers Also Bought” carousel, are still contained in static lists.

The ability to compare books is crucial when one looks at how recommenders usually provide suggestions. The most popular way is to present items similar to an “anchor book” – a term for the book on which recommendations are based. Another way is for sites to capitalize on user information like reading history and preferences; however, this still relies on similarity analyses between books. Because book recommenders are often black-box, users do not get any insight on how recommendations are generated in the first place. Choosing becomes harder if users do not know how the recommendations compare to the anchor book and to each other.

The plainness of recommendation lists is surprising considering books are high-dimensional entities that can be compared on a number of different metrics. Metadata like title, author, publication date, length, average rating and genres can be used to better organize and explore the list.

To mitigate the difficulty of comparing books, I developed Picky-Much 1, a web-based analytic visualization of book recommendations. PickyMuch provides four features that facilitate comparisons:

• A simple list view that supports sorting on 6 metrics (title, author, publication date, average rating, popularity, length)
• A dynamic scatter plot with customizable axes that helps orient the user in the recommendation space
• A novel view for analyzing and filtering books by their genres
• A tracking system that allows users to track books of interest across the 3 views

The purpose of this project is to prompt the questions: can an enriched visualization of book recommendations help users decide what they want to read more quickly? Will it help them narrow down their options in the initial stages of their search? Can it also help them find the books that they will enjoy more?

BACKGROUND
While the research community consistently expends efforts on improving recommendation systems, the information visualization component of these systems has not seen a parallel in growth. Systems that boast advanced algorithms for recommending items, such as Netflix, still rely on simple lists to present their results.

The advantage of lists is that their simplicity reduces cognitive load [8]; moreover, lists are simple to operate and familiar to a wide range of audience. However, the downside of lists

1http://infoviz-lengapps.rhcloud.com/
is that they do not facilitate comparisons between the items they contain. Lists also occlude a lot of information that can be useful to the user’s decision-making process, such as the books’ genres or popularity.

One of the projects that tackled the issue of visualizing recommendations is TalkExplorer [8] by Verbert et al. Their goal is to allow users to find academic talks and papers using what they call relevance prospects. A relevance prospect is an avenue for finding items of interest, and TalkExplorer incorporates three of them: social, content, and personal. TalkExplorer is an interactive visualization that allows a user to explore all three relevance prospects in relation to one another simultaneously. Verbert et al.’s resulting experiment shows that allowing a user to explore results increases the user’s trust in the black-box recommender, and it also improves the chances of the user discovering new items to bookmark.

Another approach at visualizing recommendations is to use a map analogy [4]. For their project, Gansner et al. used the data of 1000 TV shows, and they used a map to compare them. Shows similar to each other appear closer on the map, and shows that belong to a single group or genre are contained in a geographical unit like country or island. In the user-mode for recommendations, they use a heat map to show which areas of the map are more likely to have items that the user can enjoy based on their preferences. While this visualization shows which items are similar to each other, it does not give the user any insight on the basis of that similarity.

An area of research that is relevant to my project is visualizing digital libraries. One point of interest in this research area is to bring the experience of browsing physical collections of books to the digital space [2]. This problem is addressed in Dushay’s Virtual Book Spine Viewer [2]. The Virtual Book Spine Viewer presents a scatter plot of books, represented by a graphical icon of their book spines, within a certain category. The user can choose what characteristic of the books appear on the scatter plot’s axes. When clicking on a plot point, the visualization shows more information about the book on a right-hand panel. The scatter plot helps users visualize how books compare to one another, mimicking the act of browsing physical shelves.

The Bohemian Bookshelf [7] by Thudt et al. is another project that deals with visualizing digital libraries. It is a prototype of a visualization whose aim is to facilitate serendipitous discoveries of books. Like Virtual Book Spine Viewer, the Bohemian Bookshelf visualizes a collection of digital representations of books. However, the Bohemian Bookshelf provides not a single view, but five different ones corresponding to different design ideas that support a serendipitous experience.

PickyMuch incorporates many of these designs to support the visualization and comparison of recommended items, although PickyMuch differs from these previous examples in several ways. While both PickyMuch and TalkExplorer facilitate exploration of results from black-box systems, TalkExplorer focuses on combining three relevance prospects; on the other hand, PickyMuch focuses on comparing items based on metadata. In addition, while TalkExplorer requires that users provide information about themselves so that the system can incorporate their preferences and bookmarks, PickyMuch does not rely on personal information at all. This is important since personal information is not easily accessible on certain platforms due to privacy concerns. Most book recommendation websites do not require users to sign in to receive recommendations based on anchor books. PickyMuch aims to visualize and compare items on an impersonal recommendation list, meaning that anyone can gain some insight on the relationships of the items regardless of personal preferences.

In this manner, PickyMuch is a lot closer to the visualization goals of Virtual Book Spine Viewer and the Bohemian Bookshelf. I adapt certain design lessons from these projects to a much smaller collection since most recommendation lists only have about 15 to 50 items. Like Virtual Book Spine Viewer, PickyMuch follows the overview-plus-details workflow [9] for visualizing a large amount of data. Like the Bohemian Bookshelf, I provide multiple views to the user so that they have several different perspectives of the collection. Furthermore, PickyMuch’s novel genre grid, which is inspired by Florman’s Chart Board [3], allows users to compare books based on genre composition. To my knowledge, this is the first attempt at visualizing genre composition to compare and filter recommended items.

**PickyMuch**

PickyMuch is a web application that visualizes recommendations for particular anchor books. It follows the design of presenting an overview of the collection, then providing options to view details in differing levels of context [9]. More specifically, it provides three views, along with the functionality of tracking items across those views. The first view is a sortable list of all the items; the second is a dynamic scatter plot with multiple options for axes labels; and the third is a grid that shows the genre composition of the books. Tabs at the top of the page allow users to switch between views.
Figure 2: The scatter plot, where the user is hovering over a specific book.

**Standard View**

The most desirable characteristic of a standard recommendation list is that it is easy to comprehend and operate. Some people might prefer a clean, sleek view that does not overwhelm them with too much information. For this reason, PickyMuch has a standard view that presents the recommendations in a list as shown in Figure 1.

One functionality that I provide in this standard view which seems to be missing in other visualizations of recommendations is a sorting mechanism. The user has the option to sort the items based on these six parameters:

- alphabetically by title (default)
- alphabetically by author’s last name
- average rating with the highest rated appearing first
- publication date with the most recently published appearing first
- length of book by page count with the shortest book appearing first
- popularity as indicated by the total number of reviews, with most popular first

A sorting mechanism helps the user find books by these attributes. For example, some readers may only want to try books that have a high average rating. Sorting will allow them to concentrate on books that are highly rated.

**Scatter Plot View**

The scatter plot view, as shown in Figure 2, is inspired by Dushay’s design of using scatter plots to help users orient themselves in the collection space [2]. Scatter plots give users a sense of the distribution of the books based on their attributes. Like the Virtual Book Spine Viewer, PickyMuch allows users to customize the axes. The options for the axes are publication date, average rating, and popularity.

When users choose a new metric for an axis, the plot animates the movement of the points. This animation helps users keep the points in context as they move across the plot [1].

The jacket covers are displayed in a row above the scatter plot, and these covers are linked to the plots in the graph. If a user hovers over a single point, that point and its corresponding book is highlighted. Similarly, if a user hovers over a book, that book and its corresponding point is highlighted in the plot. I implemented the scatter plot in this manner for several reasons. First, it is not easy to incorporate the book’s image in small points on the graph; legibility is greatly reduced if one tries to fit the jacket cover into small circles. Moreover, depending on the distribution of the points in the graph, some points may be occluded. For this reason, I decided to decouple the points from the book information, preferring to have the books displayed separately at the top. This design has some additional advantages. If the user is interested in certain points on the graph, like an outlier, he or she only has to hover over that point to find its matching book. On the other hand, if the user is already interested in a book, he or she only has to hover over that book to find where it is on the scatter plot. This design allows for greater legibility without sacrificing necessary information.

**Genre Grid View**

Figure 3 shows the genre grid. Each row is made up of blocks of genres. These genres are the top ten most frequent genres Goodreads users associate with the book. The width of each block represents the proportion of that genre within the top 10. For example, if 20% of all the shelves within the top ten genres belong to Fantasy, then Fantasy will take up 20% of the entire width of the grid. Hovering over a block prompts a tooltip to pop up containing the genre’s name.

The goal of visualizing the genre composition of a book is to see what prominent genres or themes make up this book. This view was inspired by Ben Florman’s Chart Board [3] design, which succinctly visualizes the prominence of common themes at each stage of the book. Moreover, for easier association, each genre is coloured with the most compatible hue for that word, according to Lexichrome [5]. Genre composition is important information to have for those who want an idea of
how the books compare with each other. Are they made up of similar genres? If so, how prominent is each genre in each book? How many genres overlap the entire colletion space? Can a user find a book with a certain genre even if the anchor book does not have it? These are the questions that the genre grid attempts to answer.

The most powerful functionality presented in the genre view is the ability to filter books based on genre. Clicking on a genre block highlights the books that have that genre, as shown in Figure 3’s bottom image. A user can select multiple genres, and the genre grid will highlight all the books that have at least one of those selected genres. This is a powerful way for users to find and compare books based on the genres they contain.

Details Modal
In any of the three views, the user can quickly see the title and author of a book by hovering over its jacket cover. To obtain more information, the user can click on the jacket cover to prompt a pop-out modal. As shown in Figure 4, this modal contains more information about the book, such as the description, the average rating, the total number of ratings, and length in pages. In addition, the modal presents a link to the book’s Goodreads page in case the user is curious to see even more information like reviews.

Item Tracking
To ease the transition between views, PickyMuch allows users to track items across different views. If an item catches their interest in one view, they can click on the jacket cover to prompt a pop-out modal. As shown in Figure 4, this modal contains more information about the book, such as the description, the average rating, the total number of ratings, and length in pages. In addition, the modal presents a link to the book’s Goodreads page in case the user is curious to see even more information like reviews.

Currently an explicit selection via button is implemented. While an implicit selection (ie. clicking and holding shift to select multiple books) may support a more seamless integration of the tracking feature, clicking on the book covers currently serves as a prompt for the details modal. Exploring implicit selection remains for future work.

Use Case Scenario
In this simple scenario, I illustrate how someone can use PickyMuch to efficiently decide which books, if any, he or she should read next. Suppose Janet recently finished *The King of Attolia* by Megan Whalen Turner, and she really liked it. She now wants to read a book similar to it. As an avid reader of children’s Fantasy books, Janet has developed a taste for historical fantasy with a little bit of politics. Because she likes to read with her young son, she prefers to read books that do not contain a lot of romance. Janet is familiar with online book recommendation sites.

Standard View
Janet goes to PickyMuch’s page for *The King of Attolia*, where she sees the standard view of the recommendations. The large jacket covers allow legibility of the titles, and Janet finds that when she hovers over the jacket covers, a tooltip appears, bearing the title and author of each book.

Curious to see more information, Janet clicks on one of the jacket covers. A modal pops out, displaying more details about the book. Now knowing where to find more information, she closes the modal, and begins an earnest search for a new
book to read. She clicks on the sorting options and chooses “Popularity.” The container refreshes, and the books are now sorted according to popularity. Clicking on the first item, *Graceling* by Kristin Cashore, she reads the description and discovers that it is a book about two fighters. Janet notices the dark “Track This” button at the top-right corner of the modal. A tooltip tells her that she can track that item across all three views. Knowing that her son might want to read a book with lots of action, Janet decides to track the item.

The page now refreshes, and there is a large checkmark on *Graceling*. Going back to the sorting options, she now sorts by rating. She clicks on several books, and after reading some of the descriptions, she decides to also track *Crown Duel* and *The Goose Girl*.

**Scatter Plot**

Janet navigates to the next view, the scatter plot. She is not sure what information she will find on this page. The scatter plot shows mostly purple dots and three gold dots. She also notices that the jacket covers are in a list above the scatter plot, with her three tracked items appearing at the beginning. The gold dots, she realizes, are the points of the items she is tracking. Hovering over the dots, she quickly learns that doing so highlights the corresponding book on the list.

She changes the x-axis to display the average rating. The points move to their new positions through animation, which made it easier for Janet to keep track of them. There is one item that has significantly more ratings than the rest and has one of the highest average ratings. Hovering over the point, she finds that this item is *Graceling*, the first item she decided to track. She already knew that *Graceling* is the most popular item on the recommendation list; she did not know that it outshines the others by a significant amount. She now becomes more confident that *Graceling* is a good choice.

**Genre Grid**

For now, Janet leaves the scatter plot to explore the genres of the books. She goes to the genre grid, and she is confronted by long bars of different colours. Confused, Janet reads the instruction at the top, and learns what the long bars and the blocks mean. She hovers over the light purple bar and reads “Fantasy” from the tooltip. She scrolls through the page and finds that every book in the list is in the “Fantasy” genre.

Hovering over each of the genres for *The King of Attolia*, Janet was glad to see that “politics” was included in the top 10. She clicks on “politics,” and suddenly, the majority of books and bars fade in opacity. She learns that those books do not have “politics” in them. She decides to track the other books that have politics.

Janet then finds the “romance” genre, and clicks on it. Immediately, she sees that *Graceling* seems to have a lot of romance in it. As a matter of fact, “romance” is the third largest genre for *Graceling*, and it is larger compared to the romance block of other books. She realizes that this book may be something she would enjoy, but not read with her son.

Finally, Janet takes note of the five items she is interested in, and moves on to Goodreads to read reviews of them.

**Comparison with Goodreads**

Goodreads provides a lot of information about each book, but when it comes to recommendations, it uses a simple list. In order to accomplish the same tasks that Janet performed, the user would first have to look through the entire list to find the item with the most number of ratings, since there is no sorting option. Similarly, she has to look through the entire list to see the highly-rated items. Unlike PickyMuch, the user cannot see the description of the books on the same page, but has to navigate to the individual item pages. She also has to do this if she wants to see the genres of each book. In the individual item pages, the genres most frequently associated with the book are displayed as text, along with the number of users who have assigned that genre. There is no graphical indication that easily presents the prominence of each genre. Finally, it is almost next to impossible to compare books based on genre, since this information is on individual item pages.

**Implementation Details**

This next section discusses the proof of concept I have implemented for PickyMuch.

**Technology**

The back-end for the proof of concept was built using Django, a web application framework for Python. The front-end frameworks used were JQuery, Handlebars (a javascript template framework), Kickstart (a CSS library), and Sass (a compilation framework for CSS). In addition, the scatter plot was created using D3.js.

**Recommendation Information**

I selected two anchor books for Pickymuch: *The King of Attolia* by Megan Whalen Turner and *The Thirteenth Tale* by Diane Setterfield. These books were chosen primarily for the wide range of their genres. The 25 recommendations for each of these two anchor books are provided by Goodreads and LibraryThing. Metadata for all of the books is provided by the Goodreads API. To ensure that PickyMuch is a fast application, I downloaded all the information into my application’s database; hence PickyMuch does not make live calls to the Goodreads API in production.

**The Complexity and Challenges of Genre**

The biggest challenge in developing PickyMuch is preparing the data for the genre grid. For each book, Goodreads provides the top 100 shelf names associated with that book, along with the number of users who used that shelf name. For example, some users will add a book to one of their custom shelves named “Fiction” or “Romance.” I derived the genres based on the shelf names. However, many of the custom shelves are for personal purposes, such as “Read,” “Currently Reading,” “Want to Read,” or “Wishlist,” etc. There are also shelves that do not provide much insight into the content of the book, like “Favorite,” which is a popular shelf name for many users. While this indicates that the book might be well-liked, it does not help prospective readers, because the term is very subjective. I labeled these shelf names with “Personal” genre, and PickyMuch disregards them in its calculations.

Because custom shelf names are not regulated, many shelf names actually fall under the same genre. For example, shelf
names like “Children,” “Kids,” or “Juvenile” may indicate that the book is appropriate for younger audiences, but it is redundant for PickyMuch to treat these categories separately. Because there is currently no tool to help conflate these terms under one genre as far as I know, I manually assigned genres to the best of my abilities.

This problem worsens when sub-genres are considered. “Epic Fantasy,” “Paranormal,” and “Urban Fantasy” all fall under the category of Fantasy, but Fantasy readers know that the reading experience for these subgenres can be very different. One fantasy-lover may really like to read Epic Fantasy, but may avoid Urban Fantasy altogether. For this reason, I treated sub-genres as different genres so that PickyMuch will be able to display these details. Again I assigned genres to the best of my abilities.

Vague shelf names are also common. For example, what does the shelf name “love” mean? Does it mean that the user loves the book or that the book has love in it? Should it be assigned the genre “personal” or “romance”? Shelf names can also allude to different genres. Does a book shelved under “ghosts” fall under “horror,” “supernatural,” or “spiritual”? For these situations, I relied mostly on the book descriptions and other shelf names to guess the appropriate genre. However, if I want to make PickyMuch scalable, it’s not pragmatic to manually assign genres, especially if there are thousands of books.

Future Work

Genre Processing

The main factor that affects the accuracy of the genre composition in the genre grid is the assignment of genre to shelf names. As mentioned in the Implementation Details section, custom shelf names created by users can sometimes be synonymous to each other, can be vague, or can belong to multiple genres at the same time. To increase the accuracy of assigning the correct genre to the shelf name, there are a few directions one can investigate.

One, create a tool that determines whether words are synonyms. Many NLP tools can already detect whether words come from the same root word, but to my knowledge, there is currently no tool to determine synonyms. One tool that seems to be heading in this direction is TagRefinery [6]. TagRefinery is a new program that allows users to clean and process tags; however, I found that this tool is not mature enough yet, as it still involves a lot of manual work to indicate synonymous terms. Moreover, we need a tool powerful enough to determine if words are synonyms in the context of literature. “Young adult” and “teenager” may not be considered as synonyms in an ordinary context, but as a target age-range for books, they are the same.

To expand on that point, we also need a classifier that determines the correct genre(s) of a given term. I will bring up the example of the word “ghost” again. Ideally, a tool should be able to see the term “ghost” and be able to give “horror,” “paranormal,” or “spiritual” as choices for genre. This is not an easy task to accomplish, because we will need domain-specific knowledge. To generate the correct genres, one needs to consult with an expert in that genre. Only once we have come to an agreement about term-genre relationships can we automate the process of assigning genres.

Better Organization Structure

One might suggest that to mitigate the problem caused by custom shelves, predefined genres should be used instead. For example, books on Amazon belong to predetermined “browse nodes” that indicate fixed genres (e.g., Books > Literature and Fiction > Genre Fiction > Gothic). However, one downside to this is that there is less nuance and granularity that might influence the user’s decision-making process.

To address the problematic nature of custom shelf names and the limitations of predefined genres, the genre grid could be reorganized. Instead of cramming the grid with both genres and specific content terms, PickyMuch could instead have dedicated views for each. The genre grid can consist of pre-defined labels such as those from Amazon, and another view can visualize more specific content labels such as “ghost” or “vampire.” This way, users can visualize genre composition without losing more specific information about the content.

Sorting in the Genre Grid

To improve the filtering capabilities of the genre grid, when users filter by genres, books should be sorted according to the size of those genres, respectively.

Colours in the Genre Grid

The colours used for each genre is based on LexiChrome, a tool that associates words with a color [5]. However, there are many words that do not appear in the LexiChrome database, and there are also many words that map to the same color. A more efficient way to assign colours to genres must be investigated, so that the assignment can be automated.

Real-Time Lookup

Because of the manual processing involved in constructing my visualizations, I downloaded the metadata for all of the 52 books currently displayed in PickyMuch. A good direction to pursue is to find a way to visualize any book and its recommendations simply by doing a title or ISBN search.

Empirical Evaluation

PickyMuch was built in an attempt to explore research questions in visualizing book recommendations. In order to verify if richer visualizations can indeed help readers find new books more efficiently, a proper evaluation must be conducted.

Conclusion

In this paper, I introduced PickyMuch, a prototype for the analytic visualization of book recommendations. PickyMuch tries to address the problem where book recommendations appear in plain lists without any mechanism for item comparison. It helps users explore the recommendation space by providing multiple views with mechanisms to sort, filter, plot and track recommended items.

REFERENCES

DOI=http://dx.doi.org/10.1145/2909132.2909255


