A Stylometry-Based Book Recommendation System

Drew Johnston

Brigham Young University, Provo, Utah, USA

Abstract. Recommendation systems are designed to connect consumers to relevant products and information by personalizing recommendations based on the consumers' previous experiences. The vast majority of recommender systems use collaborative filtering methods that make recommendations based on the preferences of other users. This is often effective, but has one major drawback: previously unrated items cannot be recommended by the system. Content-based recommender systems overcome this drawback by making suggestions based on inherent attributes of an item. This paper describes a novel approach to content-based book recommendation using stylometry to compare the writing styles found in different books. A proof of concept model as been developed and tested, but the performance is difficult to evaluate due to the small size of the library that was developed. However, the system guarantees 100% coverage across the library and behaves as it was intended.

Keywords: Book Recommendation · Stylometry · Recommender Systems.

1 Project Description and Motivation

The development of powerful and robust recommender systems is of great interest to a variety of large companies, such as Spotify, Netflix, and Amazon, and to individuals who wish to get suggestions to expand their experiences in an intelligent manner. Companies wish to provide services to consumers to help connect them to information and products that they will find helpful and relevant. Online book stores like Amazon and Barnes and Noble have long provided such services to suggest books that their users will enjoy. Before these services, libraries have provided advising services to their readers for decades [2]. Readers' preferences can be very complex and are often difficult to simplify into keywords or subject categories, but can often be illustrated well through example. A recommendation system can personalize suggestions to the consumer based on the examples they provide of their preferences. An effective recommendation system can vastly improve consumer satisfaction, expose readers to novel and thought-provoking ideas, and bolster the awareness and learning of a community.

The vast majority of current recommender systems make use of *collaborative filtering* methods. This involves basing the recommendations on the item ratings provided by other users compared to the ratings of the user profile in question.

The system stores a database of user records and requires a large amount of user rating data to effectively make suggestions to another user. It finds other users whose preferences correlate strongly with the preferences of the user in question and recommends other items, in this case books, that these similar users liked. This approach trusts in the wisdom of the masses for recommendations. Because of this, a significant drawback of collaborative filtering is that the system tends to recommend popular books much more often than other books. Furthermore, if a book doesn't have enough ratings in the system, it won't even be considered. This may be desirable for some systems, but it makes it more difficult for lesser-known authors and books to break out and gain any visibility from the system, and it perpetuates already popular titles in the cycles of recommendation.

A *content-based* recommendation system overcomes this problem by not focusing on user ratings at all. Instead of storing user data, the system's database stores inherent attributes of the desired items. Recommendations are then made by comparing similarities of the library of items in the generated feature space. Users are still characterized by their provided preferences (still often in the form of item ratings), but these ratings are compared to other ratings, only used to develop a profile in the system's feature space to compare to the items in the library. This allows for increased library coverage as books don't need to have received other user ratings to be recommended. It also allows for system explainability, as the system can point at specific features that led to given recommendations. This could increase a reader's confidence in the system and knowledge of how to use the system effectively.

In the following sections, I detail a novel method for developing a content-based recommendation system that makes suggestions using stylometry. Stylometry uses statistical analysis to characterize an author's writing style. Basing book recommendations on writing style is a unique, novel approach, and due to difficulties in feature extraction and corpus generation, all experiments performed in relation to this paper were done with a small library of books as a proof-of-concept. Writing style presents an alternative way to differentiate between books and make recommendations that may interest a wide variety of readers across many genres.

2 Relevant Prior Research

2.1 Content-Based Book Recommendation

Very little research is availabe ont he subject of content-based book recommendation. The only published paper and recommender system I found in my exploration was created by Dr. Raymond J. Mooney and his team at the University of Texas at Austin [8].

Mooney et al. created a content-based book recommending system that makes use of information extraction methods and a machine-learning algorithm for text categorization to recommend books to users. Their system, called LIBRA, extracts information from Amazon's book review website, including title,

authors, synopses, published reviews, customer comments, related authors, related titles, and subject terms for each book [8]. It should be noted that if a book failed to have any published reviews or customer comments it would not be included in the system.

LIBRA performed well on test data, achieving over 90% precision across all genres when considering the top 3 recommendations provided by the system for each user profile [8]. The recommendations are made personally for each user profile, which must be developed by the user rating a certain number of books before any recommendation can be made.

2.2 Stylometry

Historically, stylometry has been used to attribute authorship when it is questionable. Usually, this means that stylometric analysis has not often been concerned with characterizing an author's writing style so much as distinguishing that author from a small number of other authors that could potentially be responsible for the text in question. Thus, stylometric analysis frequently concentrates on uncommon quirks displayed by an author that are different from this small number of other authors, rather than stylometric features that actually characterize a document [7]. This created some difficulties in researching effective features for a recommender system as the vast majority of research is concerning differentiating between a small number of authors.

However, some research in recent years has attempted to develop features that stylometrically characterize a document. These features include n-grams, distribution of word categories, distribution of grammar rules, phrasal and causal tag percents, distribution of sentiment and connotation, average character per word, average syllable per word, average word per sentence, average sentence per paragraph, proportion of words longer than six characters, proportion of words longer than two syllables, average punctuation marks per word, and average noun, verb, adjective, and prepositional phrases per chunk [1, 5, 11].

3 Data

In order to extract stylometric features, lengthy sample text is required from each book. Due to copyright law, corpora that collect the text of modern novels cannot be developed for public use without a number of permissions. For private use in testing this system, I developed a personalized corpus of the text from 30 modern fantasy novels. Some motivation for this project came from my own personal desire to know what book to read next, so this corpus contained 15 novels that I had read previously and 15 that I had not, but that I was considering reading. This way my own experiences with the writing style of these books could be used to gauge how system performance matched up with my intuitive expectations. The books used in this restricted corpus are printed in section 5 during the system demonstration.

An attempt was made to use a book ratings dataset provided by FastML for evaluation purposes, but this did not provide effective metrics for such a small subset of books used in the system [12]. As the library expands, this data should prove an effective resouce for evaluating the efficacy of the system.

4 Method

After assembling a small corpus, I extracted stylometric features including average character per word, average syllable per word, average word per sentence, average sentence per paragraph, proportion of words longer than six characters, and average noun, verb, adjective, and prepositional phrases per chunk [5, 11]. The syllable features were extracted using the Python module Syllapy, the phrase per chunk features were extracted using the Berkley Neural Parser, and the rest of the feature were extracted with the Python module NLTK [3, 4, 6, 10]. After collecting these features in a dataframe, they were standardized by mean-shifting by column and dividing by the standard deviation by column using the Python module NumPy [9]. This way, each feature had an equal influence on Euclidean distance between two points in the feature space.

To get n recommendations based on a list of books provided, the system calculates Euclidean distance between the standardized features for each book provided and every other book in the library. These distances are added up for each book in the list provided. The books in the library that generated the smallest total distance and were not included in the list provided are then recommended to the user.

In the user interface, options are provided to give as many books as desired, to determine whether or not to include more books by the same author(s) as in the list the user provided, and to decide how many recommendations to receive. An example demonstration of this interface is presented below.

5 Results

The following is a demonstration of the current, rudimentary interface for the system and some example outputs:

What books would you like to base your recommendations on?

1 : Harry Potter and the Sorcerer's Stone
2 : Harry Potter and the Chamber of Secrets
3 : Harry Potter and the Prisoner of Azkaban
4 : Harry Potter and the Goblet of Fire
5 : Harry Potter and the Order of the Phoenix
6 : Harry Potter and the Half-Blood Prince
7 : Harry Potter and the Deathly Hallows
8 : Warbreaker
9 : The Way of Kings

10 : Steelheart
11 : The Wizard of Earthsea
12 : Elantris
13 : Dune
14 : Gardens of the Moon
15 : The Lord of the Rings: The Fellowship of the Ring
16 : The Lord of the Rings: The Two Towers
17 : The Lord of the Rings: The Return of the King
18 : The Name of the Wind
19 : The Wheel of Time: Eye of the World
20 : The Wheel of Time: The Great Hunt
21 : The Wheel of Time: The Dragon Reborn
22 : The Wheel of Time: The Shadow Rising
23 : The Wheel of Time: The Fires of Heaven
24 : The Wheel of Time: Lord of Chaos
25 : The Wheel of Time: A Crown of Swords
26 : The Wheel of Time: The Path of Daggers
27 : The Wheel of Time: Winter's Heart
28 : The Wheel of Time: Crossroads of Twilight
29 : The Wheel of Time: Knife of Dreams
30 : The Wheel of Time: Towers of Midnight

User Input: 10

Would you like to include more books by the author(s) you've selected?

User Input: Yes

How many recommendations would you like to receive?

User Input: 5

Generating recommendations...

Top Recommendations:

1: Harry Potter and the Sorcerer's Stone by J.K. Rowling
2: Harry Potter and the Prisoner of Azkaban by J.K. Rowling
3: Harry Potter and the Chamber of Secrets by J.K. Rowling
4: Warbreaker by Brandon Sanderson
5: The Wheel of Time: The Shadow Rising by Robert Jordan

As the library expands, some changes will be made to the interface, such as a searchable dropdown list from which to select books and the option to base recommendations off of an author instead of a list of books.

Because of the diminutive size of the library, many typical evaluation metrics such as precision and recall were not meaningful in testing. However, the system attained 100% coverage and an intra-list similarity score of 0.38, which indicates that the features were complex enough to effectively differentiate between the novels despite the restriction to such a specific subcategory of literature. This is promising for future expansions of the system library.

Beyond statistical metrics, the recommender system performed as expected intuitively. When provided a book directed toward young adults, as in the example above, the recommendations were first for the other young-adult books in the library. Similarly, when recommending based on high fantasy novels, it first recommended other high fantasy novels. This is consistent with the expectations that aspects of writing style uniquely present in young adult novels or high fantasy novels influenced the feature distributions of the library and thus the recommendation results.

One unique advantage afforded by this approach is that it doesn't require any interactions by other users. Collaborative filtering methods rely on user data, and even the only published content-based book recommendation system requires that a book have at least one book review available on Amazon as well as user comments. This is a key hindrance to new or lesser-known books gaining any sort of spotlight from recommender systems like this. Relying on features determined only by writing style completely avoids this issue. Futhermore, users don't have to develop an extensive profile through rating multiple books before they can receive personalized recommendations. They can simply enter a list of one or more books on which to base their recommendations and get results in real time.

As the library expands, I look forward to comparing more statistical evaluation metrics for this system to other recommender systems to find similarities and differences in performance. Hopefully, this will shed light on any holes in this stylometric approach that may be addressed.

6 Conclusion

Content-based recommender systems have some key advantages over collaborative filtering recommender systems, particularly when it comes to book recommendation. Content-based recommender systems allow for new books and books that haven't been rated by a significant number of users to still be recommended with as much likelihood as popular, well-established books. With inherent attributes as features, explainability of content-based systems is much higher than their collaborative filtering counterparts. This approach using characterizing stylometric features for recommendation offers a unique method for receiving book recommendations that is particularly attractive for readers of fiction novels.

Performance is in line with what intuition suggests would occur when basing recommendations off of writing style. When given a young adult novel, the system first recommends young adult novels from the library. Similar behavior occurs with high fantasy novels. Additionally, the low computation cost associ-

ated with the algorithm ensures that real time recommendations would still be relatively simple to calculate even after drastically increasing the library size.

The system attains 100% coverage and an intra-list similarity of 0.38, which is indicative of features complex enough to differentiate between novels even in the limited range of popular, modern, fantasy novels that have been selected. Although any performance evaluations hold little meaning due to the very small subset of books used in the proof-of-concept developed for this paper, the system performed as expected during experimentation and shows great promise for an intriguing, novel approach to book recommendation that could be implemented fully in the future.

7 Future Work

While the features selected for the model I created appear substantial enough for the purposes of book recommendation, other important elements of writing style should be explored as features, including distribution of sentiment and connotation, distribution of grammar rules, and use of idioms, metaphors, and vocabulary choices [1].

Further exploration should also be made to discover methods of adapting this approach to include degrees of importance for each book fed to the system. This would bridge the gap to learning from user rating profiles with continuous scales. It would also be of interest to explore ways to restrict the desired library by genre or plot content, e.g. "Recommend only books with magic systems set in space" or "Recommend only realistic fiction books set in medieval Europe."

The biggest obstacle to future work is the difficulty of developing a sufficiently large corpus for system deployment. While some text for a variety of novels can be found online, it can be difficult to ensure that all copyright laws are being followed. Furthermore, the differing formats of the text files found at different resources make automated feature extraction virtually intractable. With greater resources, potentially even requiring cooperation with publishing companies, a deployable system could be implemented for readers on a large scale.

References

1. Ashok, V.G., Feng, S., Choi, Y.: Success with style: Using writing style to predict the success of novels. In: Proceedings of the 2013 conference on empirical methods in natural language processing. pp. 1753–1764 (2013)
2. Baker, S.L.: Laying a firm foundation: Administrative support for readers' advisory services. Collection Building (1993)
3. Bird, S., Klein, E., Loper, E.: Natural language processing with Python: analyzing text with the natural language toolkit. " O'Reilly Media, Inc." (2009)
4. Holtzscher, M.: Syllapy (2018), https://github.com/mholtzscher/syllapy
5. Khosmood, F., Levinson, R.A.: Automatic natural language style classification and transformation. In: BCS-IRSG Workshop on Corpus Profiling. pp. 1–11 (2008)

6. Kitaev, N., Klein, D.: Constituency parsing with a self-attentive encoder. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Association for Computational Linguistics, Melbourne, Australia (July 2018)
7. Klaussner, C., Nerbonne, J., Çöltekin, Ç.: Finding characteristic features in stylometric analysis. Digital Scholarship in the Humanities **30**(suppl_1), i114–i129 (2015)
8. Mooney, R.J., Roy, L.: Content-based book recommending using learning for text categorization. In: Proceedings of the fifth ACM conference on Digital libraries. pp. 195–204 (2000)
9. Oliphant, T.E.: A guide to NumPy, vol. 1. Trelgol Publishing USA (2006)
10. Python Software Foundation: Python language reference, version 3.6.8 (1991), https://www.python.org, available at https://www.python.org
11. Stamatatos, E., Fakotakis, N., Kokkinakis, G.: Automatic text categorization in terms of genre and author. Computational linguistics **26**(4), 471–495 (2000)
12. Zajac, Z.: Goodbooks-10k: a new dataset for book recommendations. FastML (2017)