SMU Data Science Review

Volume 1 | Number 3 Article 8

2018

Text Enhanced Recommendation System Model Based on Yelp Reviews

Peter Kouvaris
Southern Methodist University, pkouvaris@smu.edu

Ekaterina Pirogova
Southern Methodist University, epirogova@mail.smu.edu

Hari Sanadhya
Southern Methodist University, hsanadhya@mail.smu.edu

Albert Asuncion
Southern Methodist University, aasuncion@smu.edu

Arun Rajagopal
arajagopal@affinitysolutions.com

Text Enhanced Recommendation System Model Based on Yelp Reviews

Albert Asuncion[1], Peter Kouvaris[1],
Ekaterina Pirogova[1], Hari Narayan Sanadhya[1]
Arun Rajagopal[2]

[1] Master of Science in Data Science,
Southern Methodist University, Dallas, Texas, 75275, USA
{aasuncion, pkouvaris, epirogova, hsanadhya}@mail.smu.edu

[2] Affinity Solutions, 875 6th Ave 21st Floor, New York, NY 10001
arajagopal@affinitysolutions.com

Abstract. In this paper, we introduce a useful natural language model for improving recommendation systems using collaborative filtering algorithms on ordinal ratings data. Since their inception, recommendation systems have evolved from simple user-business-rating matrices to complex systems that can consume multiple dimensions. Using Yelp's competition data set, we explore extending these dimensions to include natural language by leveraging a dual neural network architecture to produce a new and improved star rating system which offers potential improvements to collaborative filtering based recommendation systems.

1 Introduction

Recommendation has become one of the most important features in technology products. Recommendation allows almost any product to be tailored to a specific user, providing empowerment without requiring excessive investment of time. Learning what users like is a key focus of many systems nowadays, exemplified when creating a new Twitter account and selecting the topics one is interested in. This new age focuses on more advanced systems that are able to know our preferences as soon as we create our account. These systems ingest our public profiles and learn what is best for us in some given context. Recommendation systems were not always this important however. They were initially limited to e-commerce systems [1], where a system recommended a small set of products based on what similar users had purchased. Now they provide movie recommendations, restaurant suggestions, and purchase recommendations, keeping hundreds of millions of users happy and engaged. They have become a necessary inclusion for any product with extensive listings, replacing manual search with a form of content easier to digest. This allows users to focus their attention on engaging with the product instead of spending time and willpower exhaustively searching for related content. Products that make extensive use of these systems are YouTube's video suggestions, Amazon's purchase recommendations, Yelp's business recommendations, Netflix's movie recommendations and

Linkedin's connection recommendations. These products are large contributors to each company's respective bottom line, and are also heavily reliant on recommendations that drive user engagement.

Revenue Tied to Reccomender Systems (2017)		
Company	**Revenue**	**Source**
Amazon	$178B	Purchases
Netflix	$11.69B	Subscriptions
YouTube	$3.96B	Advertising
Linkedin	$1.1B	Recruiting
Yelp	$0.85B	Advertising

It is necessary for recommendation driven products to leverage some form of data collection which then feeds and improves the recommendations produced, but not all data is equal when constructing recommendation systems. Extensive work has been completed on the impact different types of data can have on the overall accuracy of these systems [2]. In some cases, the product that generates data for the systems is inherently lacking in its ability to produce quality data points, and as a result recommendations trained without feature engineering are lacking.

A very common data point leveraged in these systems which is not always of great quality is the star rating. Its ubiquity is a manifestation of its utility. It is visually rich and intuitive, delivering insight at a very high data-to-pixel (or data-to-ink) ratio [3]. For this reason, the star rating is pervasive in desktop and mobile applications, able to present a spectrum of negative to positive opinions on a product or service. Anyone can appreciate a 1 star as a highly negative opinion and 5 stars as highly positive. The most common implementation is an ordinal rating system that typically ranges from 1 to 5 stars. Because it is subjective and simple to assign, star ratings tend to be ambiguous, leaving users without context to understand what the ratings mean. A 5 star rating for one user can mean something absolutely different for another. In other words, if a measure of equality, also called parity, can be established between a user's preferences and the reviewer's, then the star rating would make more sense.

Establishing parity in star ratings does not seem immediately necessary from an algorithmic standpoint if star ratings are considered having a reasonable variance. Certain methods,like the Pearson coefficient, are able to eliminate the need for parity [4]by factoring out major differences between users' interpretations of each ordinal star value. For this reason, star ratings in online reviews are not assumed to require parity for an accurate recommendation system to be built. This however is rarely the case, due to both the ambiguity that is often rampant in star rating systems and lacking density in star ratings for each user or item within a data set[5]. Our suggested alternative is a solution that feature engineers adjusted star ratings using pre-existing user data to avoid these pitfalls.

We call this less ambiguous star rating a ”super star” and it can be deployed as an alternate star rating for online reviews.

The remainder of this paper proceeds as follows: Section 2 gives a background of Yelp and a description of its star rating system and recommender model. Section 3 outlines the past, current, and future state of recommender algorithms. Section 4 outlines the key data requirements for successful recommendation engines. Section 5 introduces the reader to the set of problems Yelp's ordinal rating system has. Section 6 discusses the methods for developing the extracted information from user and business reviews using Natural Language Processing (NLP) and neural networks. Section 7 and 8 review the specifics of our implementations within this framework. Section 9 outlines ethical issues related to online reviews. Finally, Section 10 concludes by summarizing the relevance of our study on future recommendation systems.

2 Yelp Background

We have selected Yelp's[3] star ratings and recommendations from a publicly available dataset[4] due to its availability and representative data quality. Yelp has designed automated recommendation software that attempts to transform user reviews into valuable feedback via recommendations on both the business level and the review level. Yelp's recommendation software looks at dozens of different signals, including various measures of quality, reliability, and activity generated by its users. Leveraging this data, Yelp outputs dynamic content for each user to increase engagement and the value of its platform.

Yelp's product is a popular platform offering a way to introduce businesses to customers and customers to businesses. Yelp relies upon business reviews to generate advertising, which made up about 91% of the company's total revenue in 2017 [6]. The company collects business reviews from registered users averaging all registered ratings to assign a score for a given business. The product makes heavy use of recommendations by weighing reviews and returning subsets of businesses a given user is likely to find useful. The goal of recommendations is to increase user value and usage, increasing the value of Yelp's advertising inventory.

Yelp uses many features like location, context, time and user search history to provide the best recommendation for a user.[7] However, it fails to take into account the abundance of information users provide in the text review of the business. This information contains user preferences, revealing insight into important features the user pays attention to. Our goal is to leverage the text information from reviews to create a recommendation system of the future, which can highlight the benefits of extending to new sources of user preferences. As a result, we qualify user ratings and achieve parity in the star rating. Our focus when achieving this is to improve the data characteristics of the star rating.

[3] https://www.yelp.com/
[4] https://www.yelp.com/dataset/challenge

3 Recommendation Systems

The methods in which recommendation systems are built and applied are currently in an age of rapid evolution. The fundamental ideas and mathematics were introduced in the mid-1990s to help users select the most suitable product when given a plethora of options. The key idea that led to their initial development was that the opinions of the masses would be a reliable filter to be used when delivering results for an individual. In their simplest form, these algorithms take a set of observations that include a user, item, and rating, and then leverage mathematics to create a matrix of relationships that can be searched to produce a list of new items that a given user may also rate highly. This past layer of the recommendation system creation has remained relatively unchanged, with minor additions for the mathematical variations being added. One of the most popular implementations of this layer is the collaborative filtering algorithm. This algorithm builds on the assumption that peers with similar ratings and behavior in the past will also have comparable preferences in the future. We will focus on this algorithm due to its popularity in research [8], usefulness in dealing with ordinal data [9], flexibility in being expanded on [10], and the scalable options for these algorithms made available by Apache Spark.

With the past decade's explosion in computational power, data volume, and algorithmic development, recommendation methods have begun to evolve [11]. Modern systems that can consume additional dimensions like location are being applied in different ways to the recommendation stack, allowing increases in the utility provided to the user by the recommendation system. These newer forms of user data are often called implicit data, which can be compared to the rating metric often called explicit data. The collaborative filtering algorithms discussed above are still used in generating the final matrices that allow systems to deliver a user's result, but additions to these architectures are always being expanded.

We consider the new generation of recommendation systems to include additions that treat users and items as sets of networks with many possible features to consider. These networks are being described as ontologies in popular literature [11]. Separate from just a user's past ratings, new systems can consider friend networks, location, text data, images, and any other form of data analyzable with data science.

4 Key Data Requirements in Recommendation Systems

The quality of a recommendation system can be heavily impacted by the data passed to it during training. Previous research [2] [12] has explored the relative importance of certain data characteristics when building a model. These characteristics can be separated into two major groups, namely those out of control of the modeler and those within the control of the modeler. The borders around these two groups are constantly being redrawn as further research is added to the field. Understanding where limitations lie and defining areas for possible improvement is key to the facilitation of recommendation system improvement.

Data characteristics that are out of the control of the modeler can create large problems and place a ceiling on potential accuracy. A larger portion of literature in the field [13] has focused on exploring methods that can move problems originally thought out of a modeler's control back into a space where there is control. While development has been made on issues like data sparsity, which is considered the most influential data characteristic on model performance [13], there still exists a large portion of data characteristics that remain out of a modeler's control. Business level and user level variance are ranked as the second most influential data characteristics when using collaborative filtering [12] and as such is a focus of our improved super star ratings. For clarity, we have defined important data characteristics for collaborative filtering success using the terms data quality, data validation, and data volume. These characteristics are generally established by the product or system that initially captures the data. We formally define these characteristics as:

Data Quality - When there exists a beneficial amount of variance represented in standard ratings data on both the overall level and user level.

Data Volume - When there exists an amount of data that allows for an appropriate density of ratings across a majority of items in the population.

Data Validation - When there exists an accurate flagging mechanism in the data that limits items used in the system to high quality items.

Products like Amazon, YouTube, LinkedIn and Yelp all aim to solve different user problems. The result of this is that these systems each capture different forms of data. The difference in these data drastically shifts the accuracy achievable by collaborative filtering algorithms because there may be too much variance or lacking density in ratings. Considering transformations and feature engineering that can solve these problems for any given system greatly improves the accuracy achievable and as a result potential user happiness.

5 Ratings Problem

We have noted how ratings data can be noisy, imprecise, or outdated. Over time, user preferences and experience change. When a new user provides a rating of 5 to any business, it may be possible that the rating was a result of the user's limited exposure to other businesses. With time, the user rating for the same business may change. A rating of 1 for a business by a user may be a rating of 2 after a year because during that time, he may have had additional bad experiences at other related businesses. These edge cases in ratings systems can cause data quality problems which have few known solutions [10].

For Yelp, figure 1 shows the distribution of the ratings posted by Yelpers. Almost half of the reviews were 5 star rated. Around 70% were either 5 star or 4 star reviews. So when using this data, rating of the businesses is computed as the average of all the ratings for the business, its highly likely that many mediocre restaurants will have good scores.

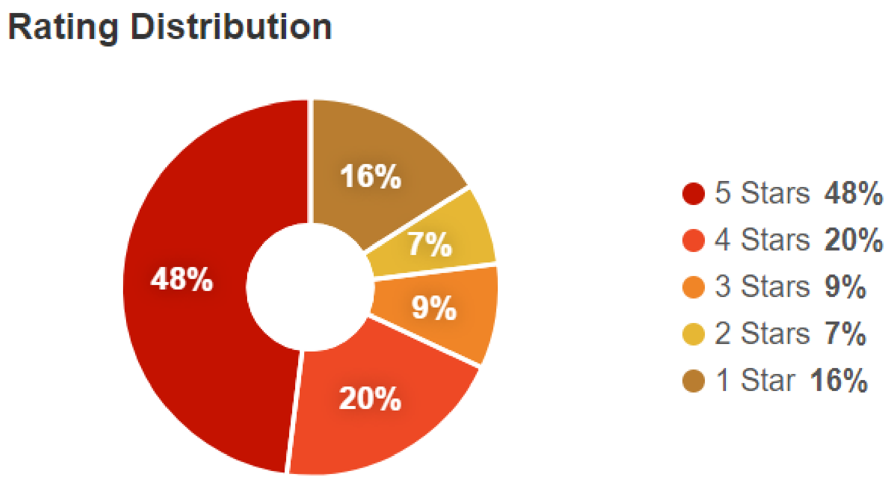

Fig. 1. Yelp Rating Distribution Rating distribution of all the rating provided by Yelp users

Ideally the rating distribution for a representative system would be a normal curve with most rating concentrated in the average experience. Only the excellent restaurants should have good scores and really bad restaurants should have score of one. Some of the restaurants should be excellent and some would be bad and most of them in the middle in the mediocre category.

6 A Primer on Neural Networks for Natural Language Processing

Leveraging neural networks for natural language processing offers a set of robust tools for transforming the high dimensional data hidden in text to something usable in a model. A neural network is structured similar to the brain's neural network. A neuron in a neural network is a simple mathematical function capturing and organizing information according to an architecture. The network closely resembles statistical methods such as curve fitting and regression analysis. A network consists of layers of interconnected nodes. Each node, called a perceptron, resembles a multiple linear regression function. The perceptron feeds the signal generated by this multiple linear regression into an activation function that may be nonlinear. In a multi-layered perceptron (MLP), perceptrons are arranged in interconnected layers. The input layer receives input patterns. The output layer contains classifications or output signals to which input patterns may map.[14] The result is a mathematical tool able to learn non-linear functions from data.

In our research we leverage the convolutional neural network(CNN), which are several layers of convolutions with nonlinear activation functions like ReLU or tanh applied to the results. CNNs are mostly used with image classification because it works when relevant structural information is present in the input features supplied to the network. What makes CNNs strong for image processing also translates over to creating powerful features from text. CNNs are most useful when there are a lot of features like pixels in an image or words in a vocabulary. CNNs are able to achieve useful learning potential with far fewer parameters. As a result, training the model is faster and requires less data as well. These technical nuances allow our dual-network architecture to more accurately sort and value the natural language text data present in our reviews.

A key aspect in achieving high accuracy using CNNs for NLP is selecting a useful preprocessing method to transform raw text data to some numeric form consumable by the algorithm described above. To achieve this we use the pre-trained 6B global vectors for word representation (GloVe) [5] embeddings made available online. Leveraging these word embeddings allow us to represent the text data in a more abstract mathematical way that is more representative than we could achieve using just Yelp's text data. The GloVe embeddings are the results of an unsupervised algorithm trained on large corpus of text. These embeddings assign mathematical values to the English words in our Yelp data set based on their complex mathematical relationship to all other words in the pre-trained corpus. The result is a set of numeric features that represents our language which can be passed to our CNN.

7 Star Creation Methods

As previously noted, the importance of having low variance within star rating systems and having a dynamic modeling method that can adjust as users do are key features in a robust recommendation solution. The end result should be a measurable improvement in the recommendation model accuracy which is robust enough to work as the underlying user behaviors evolve. To achieve this we compare a default baseline method using collaborative filtering with a simplistic sentiment model and finally a language powered cooperative convolutional neural network. Representing these tiers of results highlights an important nuance of applying transformations to star rating data. While certain methods may solve the overall non-normality of an ordinal star rating system, if the method does not apply transformation on the user and item level, then the variance still exists and will disrupt recommendations. The goal of an applied method is to reduce variance overall, normalizing the star rating so that results are more in line with what are considered optimized [2].

7.1 Default Star Baseline Model

Our baseline recommendation model takes the default star ratings as they are and uses the user, item, and rating matrices to construct a recommendation model using collaborative filtering. This baseline highlights the weaknesses inherent in Yelp's ordinal star rating data set. Large variance exists within a

[5] https://nlp.stanford.edu/projects/glove/

substantial portion of our user and business matrices. These data characteristics are influential factors when it comes to recommendation performance [12].

7.2 Sentiment Enhanced Model

Sentiment is a simple NLP method applied to text which yields a quantitative measure where lower implies negative sentiment and higher implies positive sentiment. Users leverage lexicons of words in a given language to dictate how items are scored. During our research we tested a simple replacement of star ratings with positive and negative sentiment. This method highlights only a single, yet very popular, dimension available in text. This simple NLP method is a worthy inclusion in our research, as it highlights the importance of normalizing language within a context of ratings instead of simply within the context of language. Sentiment analysis achieves the later of the points, generating metrics based on the language itself and unrelated to the business problem.

7.3 Natural Language CNN Model

Our suggested optimized architecture for solving the problems with ordinal star ratings is a Deep Cooperative Neural Network (DeepCoNN) which includes a set of independent convolutional neural networks that are coupled together in the last layer, resulting in a single output instead of two different outputs. One of these networks focuses on learning user behaviors by exploiting reviews written by the user. The second network learns business properties from the reviews written for the business.[15]. Thinking about each network independently is a key feature of this architecture.

The first neural network tries to predict the rating of the review based on:

1. A combined list of all the review texts written by the user irrespective of the business.
2. The current review text written by the user.

The primary idea here is, given the matrix showing how the user talks (all reviews) and how the user is currently talking (current review text), ask the neural network to predict the rating of the review.

The second neural network tries to predict the rating of the review based on:

1. A combined list of all the review texts written for that business irrespective of the users writing the review.
2. The current review text written for the business.

The primary idea here is, given the matrix showing how the users usually talk about the current business (all reviews) and how the user is currently talking about the business (current review text), ask the neural network to predict the rating of the review.

We have incorporated DeepCoNN using three different Neural Network algorithms namely Convolutional Neural Network (CNN), Long Short Term Memory network (LSTM) and Gated Recurrent Units (GRU). DeepCoNNs with our tested architectures are the ideal architectures for this problem space, where each method considers and mathematically weights an important piece of language, including both word order and word usage relative to other instances. Considering these factors within an architecture is necessary for producing a method that properly applies variance reduction on the user level.

8 Results

Comparing quantitative measures popular in the space, we can verify what sorts of improvements our enhanced star method delivers over the baseline and sentiment models. These include metrics related to the adjusted stars that benefit recommendation model training [12] and mathematical measures of the trained recommendation model. For measuring recommendation model success we compare root mean squared error (RMSE), mean absolute error (MAE), and fraction of concordant pairs (FCP) across the three methods. Readers may be familiar with the first two of the measures, which are simply values for the error of predicting incorrectly during regression. FCP is a common algorithm in the space which leverages the distance between predictions to generate a value between 0 and 1 where higher is better.

8.1 Star Creation Results

The improvements to star creation are most easily compared when visualized side by side. Readers will note that the initial star rating system, represented in the left-most plots, is the heavily skewed ordinal system reviewed above. Not only is the overall star distribution lacking, but overall user skewness is very high. Applying sentiment improves the former system by normalizing overall star ratings, yet does very little to adjust the average user's star distribution and as a result variance. When applying the final star method, detailed in our section above, readers can note that normalization of the overall star rating system is paired with a reduction on the overall skewness of user star ratings. This highlights an important shift towards data characteristics more valuable for collaborative filtering methods [12].

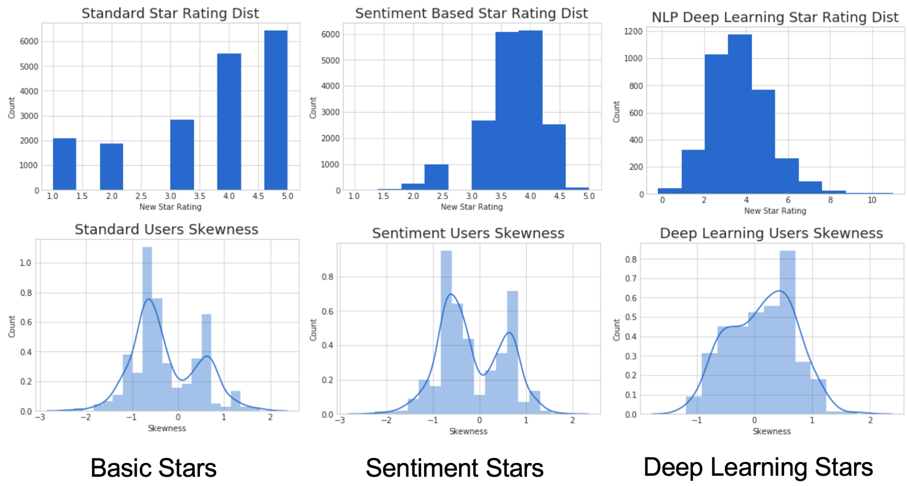

Fig. 2. Comparing Methods Comparative plots of overall user rating variance (top) and user rating variance skew (bottom) across our 3 tested methods.

Another important view when validating stars is on the business variance level. Reduced overall business variance highlights an improvement to a material data characteristic. Having higher overall business variance increases the percentage of businesses the collaborative filtering algorithm will have difficulty with, reducing the algorithms overall confidence in generating predicted scores for those items. The improved super star method greatly reduces overall business variance by a substantial margin.

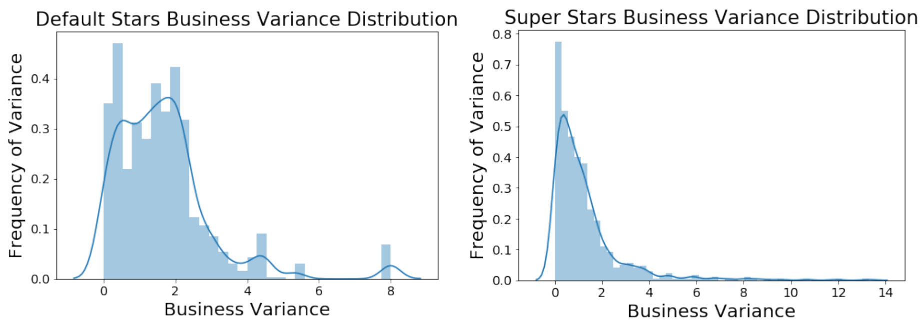

Fig. 3. Improving Business Variance Comparative plots of overall business rating variance (left) and adjusted business rating variance (right).

8.2 Recommendation System Results

As expected, our deep learning architecture achieves the highest accuracy across the three major metrics in the field. [9] Also notable, the sentiment model performs poorly on most methods, only scoring highly when it is certain of sentiment. This highlights an issue with simplistic sentiment methods where most items are given a neutral rating that does not provide a large amount of useful information. Our deep learning method is scored on out of sample data, reducing the chance it has over fit on data trained excessively.

Comparing Metrics Across Star Methods			
Metric	RMSE	MAE	FCP
Default	3.5002	3.1455	0.5439
Sentiment	7.2618	6.3695	0.5502
NLP CNN	1.2199	0.9405	0.6080

9 Ethical Problems of Recommendation Systems

The architects of economically valuable recommendation systems are ethically required to defend against attackers of their systems that may steal or destroy value otherwise earned by the system's users. Users are both businesses and normal browsing users in the case of these systems. This ethical consideration is necessary for recommendation systems to be reliable and trustworthy. Without this security, users have an economical disincentive to invest both time and money into trusting them. Where economic gain is available, hostile users will always attempt to adjust the ratings of certain businesses, both up and down.

Many attacks in this area are designed to make the recommendation system to work poorly. Over time, more attacks of this type have been developed. A popular variation uses fake user profiles to influence average. However, the integration of semantic information greatly reduced the bias injected by segment attacks. [16] This way we can infer that semantic information, as well as a text of reviews in combination with identification of fake profiles (which are already implemented by Yelp) can greatly reduce the severity of the attack on recommendation systems.

However, the challenges of recommendation systems don't end with attack mitigation alone. Recommendation systems must protect the privacy of a user. Malicious persons can easily locate a user based on the locations of the businesses the user has reviewed. In addition, businesses can pose a threat as well in the form of lawsuits for negative reviews from users. On the other hand, businesses are subject to a form of blackmail to protect its reputation in exchange for non-negative reviews. Clearly, there are threats to all Yelp stakeholders which recommendation systems must mitigate.

Adding more features to the recommender model may yield more relevant recommendations but these features may support attack methods mentioned above. User age, ethnicity, social network, health and dietary restrictions are features that would improve the accuracy of recommendations. These are data that are personal and should raise concerns for privacy. The overall goal of these systems, as we have defined, is to provide user value, however blindly providing value creates a robust set of ethical issues and weaknesses. The sweet spot is to generate a system that considers attackers methods and adjusts over time. For this to be achieved, data scientists cannot rely on one specific method or model, but instead must slowly adjust over time based on how the environment changes.

10 Conclusions and Future Work

Our proposed architecture provides a dynamic solution to many of the problems plaguing ordinal star rating systems. It adjusts overall and user level variance, increasing model accuracy. These methods also shifts ratings with changing user sentiment, accommodating for preference shift on the user level which can be expected in production systems. It greatly outperforms the alternatives along these dimensions and as such should be considered a reasonable starting point for production methods that wish to deliver user value. When contrasting this method with pre-existing star rating methods, our model leverages excessive computational resources compared to lightweight collaborative filtering algorithms. Considering the computational and knowledge worker costs of these methods are necessary for future work and implementations.

Extending this analysis to additional open source recommender data sets with ordinal rating systems and text data would increase validation of our methods. Validation in these domains should focus on the same data characteristics focused on here, namely improvements to user and item variances as well as improvements to recommendation system accuracy. A DeepCon architecture is recommended and should highlight the variance reduction available given a set of text data. Text data transformation is the key data characteristic that allows this method to provide additional recommendation model value and should be considered in future research related to this paper.

References

[1] J. Ben Schafer, Joseph Konstan, and John Riedl. "Recommender Systems in e-Commerce". In: *Proceedings of the 1st ACM Conference on Electronic Commerce*. EC '99. Denver, Colorado, USA: ACM, 1999, pp. 158–166. ISBN: 1-58113-176-3. DOI: 10.1145/336992.337035. URL: http://doi.acm.org/10.1145/336992.337035.

[2] Oren Sar Shalom et al. "Data Quality Matters in Recommender Systems". In: *Proceedings of the 9th ACM Conference on Recommender Systems.* RecSys '15. Vienna, Austria: ACM, 2015, pp. 257–260. ISBN: 978-1-4503-3692-5. DOI: 10.1145/2792838.2799670. URL: http://doi.acm.org/10.1145/2792838.2799670.

[3] Edward R. Tufte. *The Visual Display of Quantitative Information.* Cheshire, CT, USA: Graphics Press, 1986. ISBN: 0-9613921-0-X.

[4] Dietmar Jannach et al. *Recommender Systems: An Introduction.* Cambridge University Press, 2010. DOI: 10.1017/CBO9780511763113.

[5] Yehuda Koren. "Factor in the neighbors: Scalable and accurate collaborative filtering". en. In: *ACM Transactions on Knowledge Discovery from Data* 4.1 (Jan. 2010), pp. 1–24. ISSN: 15564681. DOI: 10.1145/1644873.1644874. URL: http://portal.acm.org/citation.cfm?doid=1644873.1644874 (visited on 03/14/2018).

[6] Yelp Website. *Yelp Reports Fourth Quarter and Full Year 2017 Financial Results.* 2017. URL: http://www.yelp-ir.com/news-releases/news-release-details/yelp-reports-fourth-quarter-and-full-year-2017-financial-results (visited on 07/17/2018).

[7] Yelp Official Blog. *Introducing Collections: Handpicked Recommendations, Just for You.* 2018. URL: https://www.yelpblog.com/2018/05/introducing-collections-handpicked-recommendations-just-for-you (visited on 07/17/2018).

[8] Michael Fleischman and Eduard Hovy. "Recommendations Without User Preferences: A Natural Language Processing Approach". In: IUI '03 (2003), pp. 242–244. DOI: 10.1145/604045.604087. URL: https://www.cs.cmu.edu/~hovy/papers/03IUI-recommender.pdf.

[9] Yehuda Koren and Joseph Sill. "Collaborative Filtering on Ordinal User Feedback". In: *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence.* IJCAI '13. Beijing, China: AAAI Press, 2013, pp. 3022–3026. ISBN: 978-1-57735-633-2. URL: http://dl.acm.org/citation.cfm?id=2540128.2540570.

[10] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning.* http://www.deeplearningbook.org. MIT Press, 2016.

[11] M. Shvarts, M. Lobur, and Y. Stekh. "Some trends in modern recommender systems". In: (Apr. 2017), pp. 167–169. DOI: 10.1109/MEMSTECH.2017.7937559.

[12] Gediminas Adomavicius and Jingjing Zhang. "Impact of Data Characteristics on Recommender Systems Performance". In: *ACM Trans. Manage. Inf. Syst.* 3.1 (Apr. 2012), 3:1–3:17. ISSN: 2158-656X. DOI: 10.1145/2151163.2151166. URL: http://doi.acm.org/10.1145/2151163.2151166.

[13] Joeran Beel et al. "Research-paper recommender systems: a literature survey". In: *International Journal on Digital Libraries* 17.4 (Nov. 2016), pp. 305–338. ISSN: 1432-1300. DOI: 10.1007/s00799-015-0156-0. URL: https://doi.org/10.1007/s00799-015-0156-0.

[14] Investopedia. *Neural Network*. Mar. 2017. URL: https://www.investopedia.com/terms/n/neuralnetwork.asp (visited on 05/05/2018).

[15] Lei Zheng, Vahid Noroozi, and Philip S. Yu. "Joint Deep Modeling of Users and Items Using Reviews for Recommendation". In: *CoRR* abs/1701.04783 (2017). arXiv: 1701.04783. URL: http://arxiv.org/abs/1701.04783.

[16] CHAD WILLIAMS. "Toward Trustworthy Recommender Systems: An Analysis of Attack Models and Algorithm Robustness". In: (2007).

[17] Kumar Ravi and Vadlamani Ravi. "A survey on opinion mining and sentiment analysis: Tasks, approaches and applications". en. In: *Knowledge-Based Systems* 89 (Nov. 2015), pp. 14–46. ISSN: 09507051. DOI: 10.1016/j.knosys.2015.06.015. URL: http://linkinghub.elsevier.com/retrieve/pii/S0950705115002336 (visited on 02/14/2018).

[18] Srujana Merugu Thomas George. "A Scalable Collaborative Filtering Framework based on Co-clustering". PhD thesis. Texas A & M University. URL: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.113.6458&rep=rep1&type=pdf.

[19] Petr Cvengros. "Universal Recommender System". PhD thesis. Charles University in Prague, 2010. URL: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.97.5270&rep=rep1&type=pdf.

[20] Khalid Alsamara Abuleil Saleen. "Factor in the neighbors: Scalable and accurate collaborative filtering". en. In: 1 (). URL: https://pdfs.semanticscholar.org/0894/275573ccb6fdb3bd43567b14ce0d9063903f.pdf (visited on 03/14/2018).

[21] Jure Leskovec Julian McAuley. "Hidden Factors and Hidden Topics: Understanding Rating Dimensions with Review Text". PhD thesis. Stanford University.

[22] surprise.readthedocs.io. In: (). URL: http://surprise.readthedocs.io/en/stable/matrix_factorization.html#surprise.prediction_algorithms.matrix_factorization.SVD.

[23] Justin Kuang Jeff Han and Derek Lim. In: (2014). URL: https://pdfs.semanticscholar.org/03ed/da2576f711c726682f11a8a3be7ebf42a350.pdf.

[24] Rahul Singh Richa Sharma. *Evolution of Recommender Systems from Ancient Times to Modern Era: A Survey*. URL: https://www.researchgate.net/publication/303953909_Evolution_of_Recommender_Systems_from_Ancient_Times_to_Modern_Era_A_Survey (visited on 07/17/2018).

[25] Jonathan Hui. *RNN-LSTM-GRU*. Accessed on 2018-05-05. Mar. 2017. URL: https://jhui.github.io/2017/03/15/RNN-LSTM-GRU/.