COMPATIBLE ITEM RECOMMENDATION

An Undergraduate Research Scholars Thesis

by

KEVIN J NGUYEN and VICTORIA WEI

Submitted to the Undergraduate Research Scholars program at
Texas A&M University
in partial fulfillment of the requirements for the designation as an

UNDERGRADUATE RESEARCH SCHOLAR

Approved by Research Advisor: Dr. James Caverlee

May 2018

Major: Computer Science

TABLE OF CONTENTS

Page

ABSTRACT 1

DEDICATION 3

ACKNOWLEDGEMENTS 4

CONTRIBUTORS 5

1. INTRODUCTION 6

1.1 Current Recommender Systems 7
1.2 Compatibility Recommender System 9

2. RELATED WORK 12

2.1 Item-to-Item Recommendation 12
2.2 Metric Learning 13
2.3 Matrix Factorization 14

3. METHODS 15

3.1 Problem Statement and Research Plan 15
3.2 Defining Compatibility 15
3.3 Product Entity Classification 17
3.4 Compatibility Classification 19

4. SUMMARY AND CONCLUSIONS 21

4.1 Results 21
4.2 Analysis 22
4.3 Discussion and Conclusion 22
4.4 Challenges 23
4.5 Further Study 24

REFERENCES 25

ABSTRACT

Compatible Item Recommendation

Kevin J Nguyen and Victoria Wei
Department of Computer Science
Texas A&M University

Research Advisor: Dr. James Caverlee
Department of Computer Science and Engineering
Texas A&M University

Item recommendation is an increasingly important research topic that focuses on analyzing the relationships between products to recommend items to users based on their preferences or previous activity. These systems are used extensively in different applications varying across domains to recommend items ranging from books to music. Many companies, such as Amazon, Netflix, and Spotify, leverage recommender systems to drive further engagement and revenue by delivering value through a scalable way of personalizing content for their users.

Current recommender systems recommend items based on two factors: users and items. For example, if a user purchases a product, then the recommender system will recommend similar products based on the users' previous purchases or similar social circles. In certain domains, such as clothing and electronics, the focus of compatibility relationships between products should be analyzed and used to recommend products to offer a complementary product, not a similar one.

In our thesis, we propose a new definition of compatibility to provide a new and

improved recommender system strictly for item compatibility. Compared to traditional recommender systems, compatibility recommender systems provide more accurate item recommendations for users. Our thesis currently focuses on analyzing the compatibility relationships within top-level categories in Amazon data but can be applied to any domain where compatibility is important. In order to do so, we define a general definition of compatibility, analyze a large product dataset and map product relationships, create a model to identify compatible items, and compare our results with other models. We will be analyzing the Cell Phone Accessories category with our compatibility definition. Compared to other recommender systems, our compatibility recommender system is able to recommend compatible items at a higher accuracy and can therefore be used to provide users with a more personalized experience.

DEDICATION

This thesis is dedicated to all of the hard-working individuals in the field of Web and Distributed Information Management in Computer Science, our families, and our friends.

ACKNOWLEDGEMENTS

We would like to thank and express sincere gratitude to Yin Zhang, our graduate student advisor, and Dr. James Caverlee, our advisor, for their countless efforts and contributions to our research. Their guidance helped us tremendously on our research, and we appreciate their extreme optimism and perseverance with us throughout the year.

CONTRIBUTORS

Contributors

This work was supported by a thesis committee consisting of advisor Professor James Caverlee and graduate student Yin Zhang of the Department of Computer Science.

The data analyzed for our thesis was provided by Professor Julian McAuley from the Department of Computer Science at the University of California San Diego.

All other work conducted for the thesis was completed by the students independently.

1. INTRODUCTION

The number of options people have access to is exponentially growing: millions of songs are available on Spotify, thousands of shows and movies are streamable online, and hundreds of restaurants are nearby. Due to the massive scale of the Internet, modern society provides people with a plethora of options to choose from. In the past, people shopped at physical stores, which are limited by the size of the store. By contrast, the Internet enables access to seemingly endless resources online. Amazon, for example, has an enormous collection of products but cannot display every product to every user. Due to the increase in information availability, the problem of displaying specific information to certain users arose. This gave way to information filtering systems and, more specifically, recommender systems.

For many companies, such as Amazon, Netflix, and Spotify, recommender systems drive further engagement and revenue by delivering value through a scalable way of personalizing content for their users. Modern recommender systems follow different paradigms for recommendation such as collaborative filtering, content-based recommendations, social/demographic recommendations, and contextual recommendations. Collaborative filtering compares the preferences of different users to generate predictions for users with similar preferences. Content-based recommendation leverages user preferences and domain-specific item content to generate new recommendations. Social and demographic recommendation utilizes preferences of friends, friends of friends, and demographics of similar people to suggest items. Furthermore, contextual recommendation provides recommendations based on the user's current context. For example, if a user is searching for a new car, car advertisements would be displayed and recommended as contextual recommendations.

These paradigms typically focus on helping users discover similar items. Modern recommender systems identify and understand the relationships between the items they recommend. In order to build a recommender system, a key component is that the system must have a clear definition on the relationships of items that are similar, substitutes, or complementary to develop a system that can understand a user‘s intentions and recommend items [1].

To identify the relationships between items, this would require defining an appropriate distance or similarity measure between items or learning from training data to develop a model. Providing some metric to measure between similar items is suitable for determining an equivalence relation between items. This is to ensure that we recommend items that are considered substitutes to that item. However, a distance or similarity measure will propose issues where the compatibility between items is being considered. For example, two phone cases are similar in that they provide protection for a device and composition material, but they can be entirely different due to the devices they protect.

1.1 Current Recommender Systems

Currently, other research and industry has been aimed toward analyzing the compatibility relationships between products based on their visual appearance, textual descriptions, and ratings [2, 3, 4]. Other research has used large data sets for training and provides complex models, but they follow the standard paradigm for machine learning and metric generation:

- Collect a large dataset of related and unrelated items.
- Create a similarity function to provide distance or similarity constant.
- Train the function to determine related items are more similar than non-related.

These models provide a significant amount of information for distinguishing items that are similar and can range from topics of electronics to people [5]. The metric learning model is very flexible and powerful. However, it can ignore the details where compatibility should be considered. The current models themselves are not perfect and subject to limitations:

- Similarity is either defined through an explicit category tree (e.g. ‘find the case nearest to this phone’) and this subjects the model to noise and deficiencies in defined relations. Our model and algorithms would aim to solve this by performing recommendations without dependence on explicit relationship information.

- Model approaches are too strict in recommending different items. For example, an item cannot be compatible with itself or do not generate a diverse set of recommendations, such as recommending a similar product from a different brand. By analyzing the compatibility and relationships between products in a new and creative way, we can handle these issues.

Figure 1.1 describes an example of what ‘compatible’ items would be recommended to the user given a set of queried products with a type of metric learning model: logistic regression. Logistic regression can be defined as follows.

1.1.1 Logistic Regression

Suppose $\mathbf{f}_i$ is the features that we get from product i (which can be the concatenation of product image features, description features, and ratings). If p is the probability that the two products are compatible, then

$$logit(p) = b_0 + \beta \times \mathbf{X}_{ij}$$

where $\mathbf{X}_{ij} = |\mathbf{f}_i - \mathbf{f}_j|$.

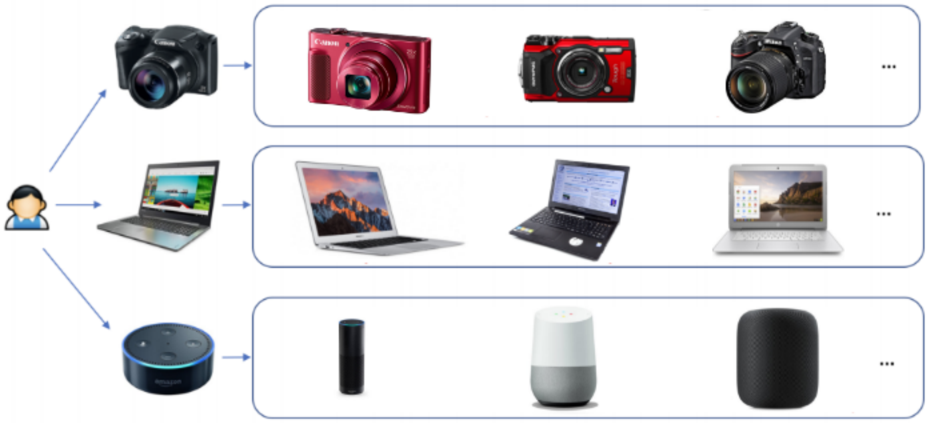

Figure 1.1: Diagram of the logistic regression model recommendations.

1.2 Compatibility Recommender System

In contrast, we are focused on discovering complementary items. For example, if a user purchases an iPhone 4, current recommender systems will recommend other similar iPhones. However, in many cases, recommending complementary items such as cases, chargers, or headphones is more relevant. Although these items are not directly similar to an iPhone 4, these items should be considered equally as important to the end user since they are additional accessories that may be needed for the device and its functions. Figure 1.2 shows an example of the compatible recommendations recommended from our compatibility classification model given the same set of queried products.

For two items to be compatible, such as a phone and a charger or a dress and shoes, they must be similar in some ways but systematically different in others. Because compatibility is a human notion that is difficult to capture through the analysis of similar item relationships, compatibility of products is usually defined manually through expert individuals or assumed through the co-purchasing habits of customers, like on Amazon. In

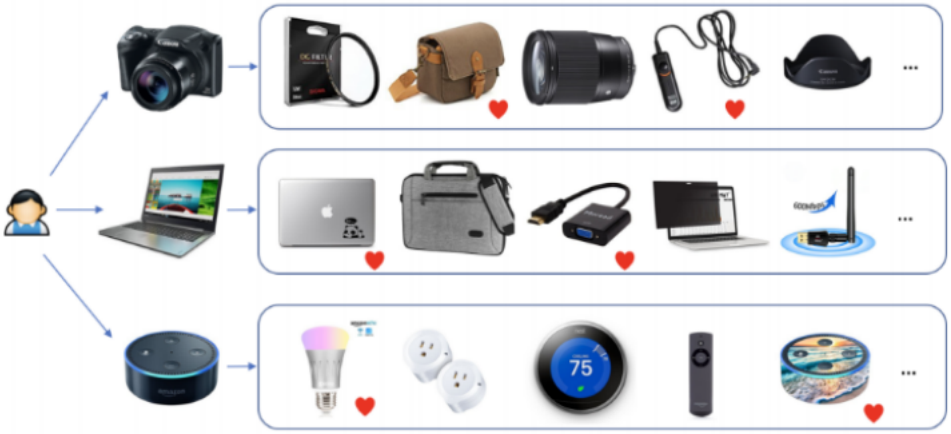

Figure 1.2: Diagram of the compatibility model recommendations.

a sample of 500,000 products in Electronics on Amazon, only 20% explicitly mention "compatibility" with another product, and therefore, assuming compatibility through co-purchasing habits can be very effective on a large scale but can be very prone to noise. For example, if a customer purchases an iPhone and an iPhone charger together, it is assumed that the iPhone and iPhone charger are compatible. However, the issue arises when a customer purchases two entirely unrelated products such as an iPhone and a t-shirt. The recommender system will now assume these two items are compatible when in reality they are not. Therefore, it is not clear how to correctly identify compatible items, especially for large and varied product sets. Another challenging aspect is finding the number of items that are compatible with the queried product among a huge dataset. Due to the large product data, randomly selecting an item has a less than 1% chance of being compatible with the desired item. Thus, our research focuses on defining a new definition of compatibility in item recommender systems to provide scalable, improved item recommendation.

While using our improved item recommendation model, not only do customers have a more personalized space to do their shopping, but sellers also have a higher chance of being recommended to users, which may increase their revenue and product views. This is phenomenon is otherwise known as the cold-start problem. The cold-start problem is the issue where new products that are placed on the marketplace do not get as much attention as the older, more reviewed products. This is because of the co-purchasing effect, where items on Amazon are recommended based on customers and their buying techniques. Because these items are new and have not been bought with other items, the chances of these items making it into recommendation systems are very slim. However, our recommender system will solve this issue by not deriving compatibility from co-purchasing items but rather from the analysis of the product and its relation to other products in more concrete ways.

2. RELATED WORK

We relate our work in the context of prior studies and implementations of (1) item-to-item recommendations, e.g. systems that generate item-to-item recommendations by analyzing the relationship between items; (2) studies involving metric learning to build relationships between different sets of items; and (3) matrix factorization.

2.1 Item-to-Item Recommendation

The analysis of the relationship among items is fundamental to modern real-world recommender systems, e.g. to generate recommendations of new songs on *Spotify*. As such, the closest systems to the compatibility recommender system we are proposing above are content-based recommender systems [6, 7] which attempt to model the user's preference toward items utilizing a variety of different features while using a similarity function, such as Pearson similarity [8], cosine-based similarity [9], and conditional probability-based similarity [10]. These systems typically analyze the metadata from the user's previous activities and content. In comparison, other recommender systems utilize collaborative recommendation approaches, e.g. counting the overlap between users who have liked both songs, as in *Spotify's* own solution [11]. This type of recommendation allows the system to recommend items based off of similar user preferences and ratings, but requires a plethora of data in order to function effectively. In addition, item recommendation systems that utilize both content-based and collaborative techniques haven't been used to address the sparsity of data available and the cold-start problem (where products are invisible to the recommender system due to newness of the item) [7]. Other approaches for item-to-item recommendation incorporate additional features, such as images for fashion recommendation and phrase-level sentiment analysis.

The methods above determine the similarity between objects. In contrast, more research has been focused on detecting relationships between items that substitute or complement one another [12]. For example, [13] focuses on the analysis of also-bought products and bought-together products to create compatibility relationships. *Murillo et al.* analyzed photos of groups of people in social media to identify which groups of people are more likely to socialize, thus providing similarity distance measure between images [14]. [15] provides more details on many of these types of recommender systems and challenges faced in this domain. Finally, [7] provides next-generation approaches on how to improve item recommendation. Our model provides a solution to many of these next-generation approaches.

Unlike content-based recommender systems or collaborative filtering, our recommender system analyzes the content of items individually in a dataset and maps each item with a compatibility relationship to another item through entities defined by our definition of compatibility. Therefore, our recommender system does not need the preferences of other users and does not require the domain knowledge that content-based recommendations are derived from.

2.2 Metric Learning

The analysis of the relationship between objects is a vast topic that covers more domains than just recommender systems. In modern learning, one is given a collection of relationships between items, and the goal is to identify a function that matches these relationships. The function must be able to generalize the relationship between objects and apply them to new unseen items to predict new relationships. The function is measured against valid data, and the metrics show how accurate the model can identify the relationships created. The most developed and advanced learning methods are used to identify hidden variables or factors among items. This can be done through matrix-factorization or collaborative filtering.

Utilizing metric learning, our main goal is to relax the model assumptions that current models have and allow for more complex notions of ‘relatedness’. There are algorithms that work with non-metric learning of relationships, but to the extent of our knowledge do not scale well with larger sets of data.

2.3 Matrix Factorization

Matrix Factorization is a concept used in recommendation systems that recommends items to users based on previous users and their rating patterns for specific items [16, 17]. Based on these rating patterns, higher positive ratings for items corresponding to users and their existing rating patterns will show a user’s interest in that item and similar items. This will allow recommendation systems to detect what users will rate future items and whether or not such items should be recommended to the user [16, 17].

Matrix Factorization is just one of the methods for recommending items. However, in the world of compatibility, matrix factorization solely is unable to solve many of the challenges that compatibility presents. Matrix factorization shows user interest based on previous sentimental ratings, but this doesn’t necessarily mean the items that the user rates positively are in any way compatible with each other. Our thesis aims to not only improve user interest but also increase existing compatibility nature between items.

3. METHODS

In this chapter, we present the problem of compatible item recommendation, and then, we present the design of our compatible item recommendation model.

3.1 Problem Statement and Research Plan

Given a set of items $I = \{I_1, I_2, ...I_{|I|}\}$, we are trying to determine for each item i in I a set of compatible items $C_i = \{I_{c_1}, I_{c_2}, ...I_{|c_i|}\}$. Specifically, we are utilizing a large real-world dataset provided by Amazon introduced in [3] which features over a million products and 42 million co-purchased relationships across 20 top-level categories. We focus on the category of Cell Phone & Accessories due to the prevalence of a variety of compatible characteristics. More specifically, we analyze all of the relationships between products in the top-level category of Cell Phone & Accessories and their subcategories in order to find the compatible set C_i for each product $i \in I$. We want to note here that although we are doing this analysis with just the Cell Phone & Accessories category, this compatibility model can be applied to all top-level categories in the Amazon dataset. To accomplish our problem statement, we separate our solution into three parts: (i) define our definition of compatibility; (ii) classify each product with a specific product entity; (iii) utilize our compatibility classification to model complex compatible relationships.

3.2 Defining Compatibility

In this section, we will define our definition of compatibility utilizing the type of relationships between top-level categories, their sub-categories, and the relationship within sub-categories. First, we organized each of the 20 top-level categories by their unique sub-categories and hand defined each sub-category's relation between each other and itself. For example, varying cases will be a size compatibility constraint, while varying chargers will

Sub-Category Compatibility	
Name	Type
Cases	Size
Bluetooth Headsets	Interconnectivity
Wired Headsets	Interconnectivity
Chargers	Interconnectivity
Cell Phone	Size & Interconnectivity
Screen Protectors	Size
Internal Batteries	Size & Interconnectivity
External Battery Packs	Interconnectivity
Battery Charger Cases	Size & Interconnectivity
Data Cables	Interconnectivity
Smart Watches and Accessories	Interconnectivity

Figure 3.1: Selected Cell Phone & Accessories Sub-Categories and Type of Compatibility

be an interconnectivity compatibility constraint. Figure 3.1 describes this in more detail. There are some examples where size and interconnectivity both play a role in defining compatibility for a specific product. For example, the sub-category battery charger cases have both a size and interconnectivity component; the case has to be compatible with the size of the phone while the charger has to be compatible with the interconnectivity of the phone. While defining these relationships, we took caution to only consider the core functionality of the sub-categories, e.g. a case must be based on size and not fashion-sense. As a result, our definition of compatibility is derived from the relationship between sub-categories within top-level categories and comes in two forms: size and interconnectivity, for a specific product.

3.2.1 Size

As part of our definition of compatibility, we define size compatibility as the relationship between product x and product y such that x and y are strictly related to each other based on their physical appearance and dimensions while requiring that x and y both have the same top-level category. However, that means that x and y do not necessarily have to be in the same sub-category. In fact, it is crucial that x and y are in differing sub-categories for compatibility to be apparent. For example, an iPhone 4 and an iPhone 4 case belongs to the same Cell Phone & Accessories top-level category. However, an iPhone 4 Case may belong to the sub-category of Cases, while an iPhone 4 may belong to the sub-category of Cell Phone. Therefore, the iPhone 4 and iPhone 4 Case have a size compatibility relationship. However, because an iPhone 4 and an iPhone 4s are both classified under the same sub-category of Cell Phone, they are not size compatible with one another.

3.2.2 Interconnectivity

Furthermore, we define interconnectivity compatibility as the relationship between product x and product y such that x and y are strictly related to each other based on their potential connectivity with each other while still requiring that x and y both have the same top-level category. However,there are many products in the same sub-category as x that may not be interconnectivity compatible with y. For example, an iPhone 4 charging cable belongs to the sub-category of Data Cables in Cell Phone & Accessories and has an interconnectivity compatibility relationship with the iPhone 4 in the Cell Phone sub-category in Cell Phone & Accessories, but an iPhone 7 in the same Cell Phone sub-category in Cell Phone & Accessories is not interconnectivity compatible with the same charging cable. To solve this, we develop a product entity classification that utilizes a natural language platform and our classification schema to build relationships between interconnectivity compatible products.

3.3 Product Entity Classification

After the classification of size and interconnectivity of sub-categories within top-level categories, we strengthen our definition of compatibility by classifying each product in

Sample Products			
	Name	Category	Entity
1	Apple iPhone 4 AT&T 16GB White	Cell Phone & Accessories > Cell Phone	Apple iPhone 4
2	iPhone 4 Hello Kitty Case	Cell Phone & Accessories > Cases	Apple iPhone 4
3	Samsung S3 USB Cable	Cell Phone & Accessories > Cables	Samsung Galaxy S3

Figure 3.2: Sample Products and their entity classification. The symbol ’>’ denotes a subcategory relationship.

each of these sub-categories with an entity. This entity will allow us to build our compatibility model and decide which products are compatible with other products.

In Figure 3.2, we see sample products with their respective entity classifications. These classifications are necessary to decide size and interconnectivity compatible items. For example, product 1 belongs to the sub-category of Cell Phone while product 3 belongs to the sub-category of Cables. These items would be considered compatible if not for the interconnectivity compatibility definition. However, because these entities are different, we classify that product 1 and product 3 are not compatible. As another example, consider product 1 and product 2. Product 1 and product 2 belong in different sub-categories under the same top-level category Cell Phone & Accessories. However, their entities are identical, classifying product 1 and product 2 to be compatible with each other. Our classification schema classifies over 340,000 products in this manner.

3.3.1 Product Entity Classification

In order to derive the product entity information, we utilized a natural language processing platform along with our classification model to recognize different entities of each product. After analyzing the benefits of multiple services, such as IBM Watson and Microsoft Cognitive Services, we decided to leverage the Google Cloud Natural Language

Platform (GCL) due to the platform's ease of use, rapid response time, and consistent results. First, we analyzed the title and description of each product and queried GCL with this result. GCL then classifies each of the products into multiple entities based on product resemblance. Entities that GCL were able to classify include CONSUMER GOOD, ORGANIZATION, PERSON, LOCATION, EVENT, and etc. We used these entities, specifically CONSUMER GOOD, to decide what type of product the queried product was. In our case, CONSUMER GOOD was the only entity that described products that were purchasable objects. Using this entity, we found multiple amounts of CONSUMER GOOD entities for each product. To be more accurate in our classification, we decided to find the CONSUMER GOOD entity with the highest salience or accuracy percentage. In this way, we are able to be more accurate with our product entity classification model in analyzing what each product actually is and what items are compatible with each product. This CONSUMER GOOD entity corresponds to the entity that is in Figure 3.2.

We mapped each product to its highest CONSUMER GOOD salient entity. With this information, we created the product entity mapping for each product.

3.4 Compatibility Classification

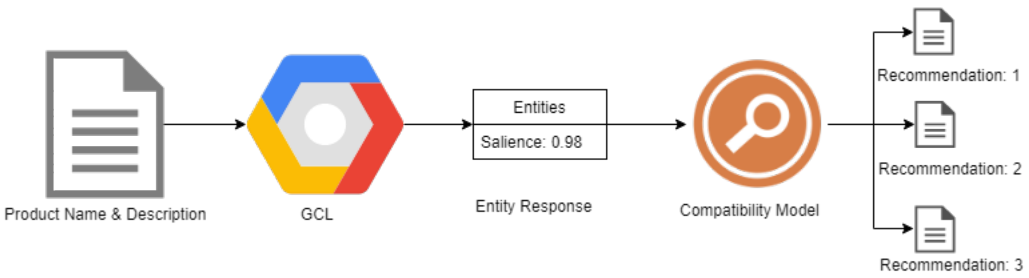

Figure 3.3: Item compatible recommendation work flow.

Utilizing the mapped product data in conjunction with the sub-category classification, we designed a compatibility classification model as follows. Each product has an entity mapping created by our classification model along with GCL. With this entity mapping, we are able to analyze compatible products, which are products that share the same entity but are in different sub-categories with the same top-level category. These are the items we would recommend the user.

New products seeking compatibility recommendations are passed through GCL by title and description to get specific entities for that product. From there, the highest CONSUMER GOOD salient entity is gathered from the list of entities and is matched to our model to find other products that are compatible with the same CONSUMER GOOD salient entity. Next, we leverage all of the other sub-categories under the same top-level category of each item that is analyzed with the matching CONSUMER GOOD salient entity. From there, we return a subset of items from each sub-category as compatible items based on interconnectivity and size. Figure 3.3 shows a pictorial representation of our model and work flow. In this way, a variety of products cross-sub-category is recommended to the user that is compatible with the original product.

4. SUMMARY AND CONCLUSIONS

4.1 Results

In order to qualitatively test our compatibility recommender system, we randomly selected 7 products from the dataset and analyzed the compatibility of the recommendations returned from four models: random selection, baseline, also-bought, and our compatibility model. For each model, we experimented with 5 recommendations for each product, and we determined if each product was compatible with the other products or not.

4.1.1 Random Selection Model

The random selection model divided the Cell Phone & Accessories into a uniform distribution, allowing each product to have an equal chance of being recommended. Due to the large dataset and variety of products offered by Amazon, we expect very few to none randomly selected compatible items.

4.1.2 Baseline Model

The baseline model is a machine learned model created with logistic regression to recommend compatible items.

4.1.3 Also Bought Model

The Amazon dataset includes metadata about purchasing information such as 'products also purchased with x'. We include this model in comparison to our compatibility model as this is the standard that Amazon uses to currently recommend compatible items to users.

4.1.4 Compatibility Model

The compatibility model utilizes our new definition of compatibility to recommend compatible items to users.

4.1.5 Data

Figure 4.1 shows the results from our experiments.

Compatible Item Recommendation Accuracy				
Product	Random	Baseline	Also-Bought	Compatibility
1	0%	0%	0%	80%
2	0%	0%	0%	0%
3	20%	60%	50%	100%
4	0%	0%	0%	0%
5	0%	0%	0%	0%
6	0%	0%	0%	0%
7	20%	0%	60%	100%
Avg	5.71%	8.57%	15.71%	40%

Figure 4.1: Compatible item recommendation accuracy with random, baseline, also-bought, and our compatibility models; tested for compatibility against 7 randomly selected products.

4.2 Analysis

Overall, our model had an overall higher average accuracy when recommending compatible items compared to the other models. More specifically, our model beats the baseline state-of-the-art model by 31.43% and the also-bought model by 24.29%. Compared to the random model, the compatibility model does significantly better (34.29%). These numbers are expected because out of all the products in the dataset, only roughly 1% are compatible with the actual product. Therefore the probability of finding compatible items with that queried product is very low.

4.3 Discussion and Conclusion

In conclusion, we utilized our definition of compatibility to create a new recommender system that leverages the human notion of compatibility to recommend items to users while capturing the complex relationship of compatibility through a mixture of metadata, natural language, and entity mappings. While there are many other existing methods for compatible item recommendation that suffer from limitations such as sparsity in review data or the cold-start problem (a recommender system cannot draw any inferences for users or items due to lack of sufficient information), our implementation allows us to avoid and relax these constraints.

4.4 Challenges

There are millions of items but only a small fraction actually mention compatibility. Our model further improves these notions of compatibility by defining the relation between products, whereas the small fraction that mention compatibility do so in a generic sense. As we have mentioned earlier, another difficulty is out of all products offered, at max only 1% of the products are compatible items with the queried product. Finding this 1% of compatible products is challenging in a dataset of millions.

Another main challenge to note here is relaxing the constraint of the cold start problem. The cold start problem is the problem that occurs when a new product is introduced. Because traditional recommender systems recommend compatible items based on co-purchasing, this new product has a very low chance of having any recommendations associated with it. This new product also has a very low chance of being recommended from other product selections. Our definition solves the cold start problem by not relying on co-purchasing of products. Therefore, the cold start problem doesn't affect how we determine or define our definition of compatibility.

Sparsity in review data is also a very common challenge faced by many other compatibility recommender systems. However, our model also relaxes this problem by not focusing our attention to users, actions, and behaviors based on the querying user but on the meaning of the product itself and not based on other users and their habits.

4.5 Further Study

Currently, we have a compatibility classification model for over 340,000 products. In order to broaden the scope of our classifier, we will look further into analyzing the relationships of compatibility for other datasets and see if these relationships can be learned through machine learning and neural methods. We can also improve our classifier by researching more in depth about natural language and creating our own natural language platform for text recognition. Finally, we could also incorporate matrix factorization along with our compatibility model to even further improve user and compatibility interests. We believe that we can do better to improve our accuracy and continue to make our definition of compatibility stronger in the future.

REFERENCES

[1] G. Linden, B. Smith, and J. York, “Amazon.com recommendations: Item-to-item collaborative filtering,” *IEEE Internet Computing*, 2003.

[2] J. McAuley, R. Pandey, and J. Leskovec, “Inferring networks of substitutable and complementary products,” *KDD*, 2015.

[3] J. McAuley, C. Targett, Q. Shi, and A. van den Hengel, “Image-based recommendations on styles and substitutes,” *SIGIR*, 2015.

[4] K. Menon and C. Elkan, “Link prediction via matrix factorization,” *ECML*, 2011.

[5] M. Der and L. Saul, “Latent coincidence analysis: A hidden variable model for distance metric learning,” *NIPS*, 2012.

[6] M. Lew, N. Sebe, C. Djeraba, and R. Jain, “Content-based multimedia information retrieval: State of the art and challenges,” *ACM TOMCCAP*, 2006.

[7] G. Adomavicius and A. Tuzhilin, “Toward the next generation of recommender systems: A survey of the state-of-the-art possible extensions,” *IEEE Transactions on Knowledge and Data Engineering, Vol. 17, No. 6*, 2005.

[8] P. Melville, R. Mooney, and R. Nagarajan, “Content-boosted collaborative filtering for improved recommendations,” *AAAI*, 2002.

[9] M. Deshpande and G. Karypis, “Item-based top-n recommendation algorithms,” *ACM Transcations on Information Systems (TOIS) 22*, 2004.

[10] G. Karypis, “Evaluation of item-based top-n recommendation algorithms,” *CIKM*, 2001.

[11] M. Mithun, “Music recommendation system spotify - collaborative filtering,” *MUS-17*, 2017.

[12] T. Li, A. Liu, and C. Huang, “A similarity scenario-based recommendation model with small disturbances for unknown items in social networks,” *IEEE Access 4*, 2016.

[13] Y. Zhang and J. Caverlee, “Neural compatible item recommendation,” 2018.

[14] A. Murillo, L. Kwak, and L. Bourdev, “Urban tribes: Analyzing group photos from a social perspective,” *CVPR Workshops*, 2012.

[15] F. Ricci, L. Rokach, and B. Shapira, *Recommender Systems Handbook.* 2015.

[16] Y. Koren, R. Bell, and C. Volinsky, “Matrix factorization techniques for recommender systems,” *IEEE Computer Society*, 2009.

[17] Y. Koren, “Factorization meets the neighborhood: a multifaceted collaborative filtering model,” *KDD*, 2008.