Finding Similar Movies: Dataset, Tools, and Methods

Hongkun Leng[1], Caleb De La Cruz Paulino[1], Momina Haider[1], Rui Lu[1],
Zhehui Zhou[1], Ole Mengshoel[1], Per-Erik Brodin[2], Julien Forgeat[2], Alvin Jude[2] ‡

Carnegie Mellon University[1]
Silicon Valley
U.S.A.
{hongkunl,cdelacru,mominah,rlu1,
zhehuiz}@andrew.cmu.edu,
ole.mengshoel@sv.cmu.edu

Ericsson Research[2]
Silicon Valley
U.S.A
{per-erik.brodin,julien.forgeat,
alvin.jude.hari.haran}@ericsson.com
(‡ corresponding author)

ABSTRACT

Recommender systems are becoming ubiquitous in online commerce as well as in video-on-demand (VOD) and music streaming services. A popular form of giving recommendations is to base them on a currently selected product (or service), and provide "More Like This," "Products Similar to This," or "People Who Bought This also Bought" functionality. These recommendations are based on similarity computations, also known as item-item similarity computations. Such computations are typically implemented by heuristic algorithms, which may not match the perceived item-item similarity of users. In contrast, we study in this paper a data-driven approach to similarity for movies using labels crowdsourced from a previous work. Specifically, we develop four similarity methods and investigate how user-contributed labels can be used to improve similarity computations to better match user perceptions in movie recommendations. These four methods were tested against the best known method with a user experiment ($n = 114$) using the MovieLens 20M dataset. Our experiment showed that all our supervised methods beat the unsupervised benchmark and the differences were both statistically and practically significant. This paper's main contributions include user evaluation of similarity methods for movies, user-contributed labels indicating movie similarities, and code for the annotation tool which can be found at http://MovieSim.org.

Keywords
Recommender Systems, Item-Item Similarity, Crowdsourcing, Supervised Learning, MovieLens.

1 INTRODUCTION

The Role of YML and MLT Recommender Systems. With the increase of online retail stores with massive offerings, users can easily get lost and suffer from information overload. Recent advances in machine learning have provided methods to assist users in these extremely large online stores. This is typically done by reducing their visible size to what is cognitively manageable by the users, by only surfacing the items most relevant to them. The most common approach to do this is with Recommender Systems (RSs). The idea behind RSs is to use past user interaction data to predict what they will want or like, and only present (or display) those items. Netflix, for example, has been quite vocal about their use of RS techniques, and have claimed that they improve user experience in general by allowing users to quickly find movies they want to watch [BL07].

We can partition RSs based on two features they can potentially provide: "You May Like" (YML) and "More Like This" (MLT). In a YML RS, users are shown a list of items they are predicted to enjoy, based on preferences they have provided for other items. MLT recommendations, on the other hand, generally surface when a user selects one specific item. Here they are usually presented with a list of details about the item; for a movie, details would include the name, description, director, year it was released, and so on. Below such details there is generally a list of other similar items, with a header such as "More Like This," "Similar to This," "You May also Like," "People Who Liked This also Liked," and so on. This presentation is analogous to walking into a physical store, going right to the item you want to purchase, and then being able to look at other similar items nearby to aid in the decision making process.

YML RSs generally rely on user consumption data in order to build machine learning (ML) models. This research area gained popularity after Netflix released a massive amount of user-to-item ratings data [BL07].

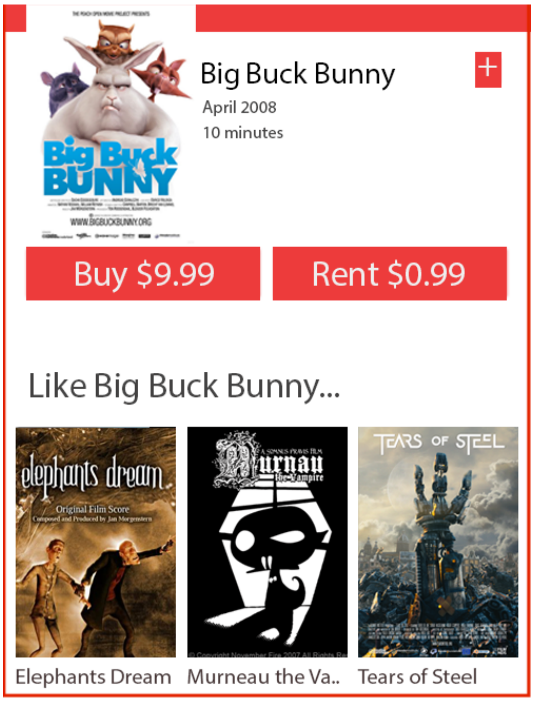


Figure 1: An example of similar movies used to perform "More Like This" recommendations. Image from [CDL16], used with permission.

After this movie dataset was retracted for legal reasons, the RS community built a replacement dataset called MovieLens [HK16]. Similar datasets have been released for other domains including music [JDE07; DWE05]. These datasets have made it possible for ML and HCI researchers to study the impact of different RS algorithms on user experience. The MovieLens website[1] even allows anyone to create an account and submit movie ratings as a way to contribute to RS research and development.

MLT has, compared to YML, received less attention. In the cases where researchers have required a means to find similar movies, they have often defined similarity heuristically based on metadata instead of taking a data-driven approach. For example, researchers have defined that two movies are similar if they share *genre*, *cast*, and *director* [BHY97].

The Need for User-Centric MLT Recommenders. We we see a major difference between how YML and MLT RSs are built up to this point. A YML RS often uses a data-driven approach while an MLT RS is often defined by the designers. This approach goes against general HCI principles including the slogan that "designers are not users." Clearly, there is an opportunity to bring in more data into MLT RSs [Nie08]. Since user interactions are increasingly shaped by ML methods and models, HCI and ML researchers need to work together to ensure that the ML methods that interface with users are evaluated and improved with the user in the centre. We believe there are two plausible reasons for the lack of focus on MLT despite its pervasive use in current-day technology: (1) There is a lack of movie-movie similarity datasets, which hinders work by ML researchers. (2) There is also a lack of validation that datasets or labels could even be useful to improve the user experience. Taken together, (1) and (2) form a vicious cycle, since we need datasets to improve user experience, and it is hard justifying a large-scale data collection activity without knowing if labels could even be helpful to users.

Our Contributions to MLT Recommenders. We hope, via this work, to break this vicious MLT cycle by collecting and releasing similarity data for movies, and by showing that data-based similarity predictions can match users' perception of similarity. All datasets and tools can be found at the project website.[2] We believe this is the first paper to demonstrate the benefit of this data-driven approach to movie similarity, having potential impact on RSs more broadly.

Our hypothesis was that the current methods used to find similar movies can be improved if we use a data-driven approach, where labelled data is used to build supervised machine learning models. These supervised methods built on user-contributed data indicating perceived similarity would better match users definition of similarity, lead to improved perceived similarity and therefore improved experience.

In this paper, we use a small dataset from Colucci et al. [CDL16] to learn four different ML models. Experimentally, we show that the four different models can be used to predict movie similarity in a way that is consistent with users' perceived similarity. Specifically, our main contributions are: (1) Empirical evidence that ML models can be used to predict similar movies in a way that more closely matches user perception than previous work. (2) An evaluation of four ML methods that can be used to build lists of similar movies. (3) A novel dataset containing almost 13,000 labels and intermediary data required to build a list of similar items for movies in MovieLens 20M.

The Structure of this Paper. We discuss related work in Section 2. In Section 3 we present machine learning methods used to learn similarity models from data. The design of the user test to evaluate those similarity models is laid out in Section 4, while the results of the study are presented in Section 5. The paper is wrapped up with discussions in Section 6 and conclusions in Section 7.

[1] https://movielens.org

[2] http://MovieSim.org

2 RELATED WORK

The first question we asked was "what makes people believe two items are *similar*?" Recent advancement in psychology and cognitive science support the notion that people use a dual-process model, whereby perceptions of similarity is built on a combination of feature-based taxonomic relations, and relationship-based thematic relations [WB99]. Taxonomic or hierarchical relations are based on internal characteristics, such as features of the items themselves, while thematic relations are external; there is a separate event or scene that connects the two items. For example, cars and motorcycles are taxonomically similar since they share many features; both have engines, wheels, and fall under the category of "ground transportation". Motorcycles and helmets are thematically similar since they are often used during the same event, i.e. a person riding a motorcycle [EGG12]. Individuals appear to favour either thematic or taxonomic similarity, and at varying levels, and with an individual's preference remaining the same even across different concepts [MG12].

Similarity algorithms (or "methods") are generally built on the intuition that "two objects are similar if they are referenced by similar objects" [JW02]. Two common methods are item-item collaborative filtering (I-I CF) and content-based (CB) similarity. In I-I CF, items are considered similar based on their relationship to users. E.g. two movies would be considered similar if they are both watched or similarly rated by a similar group of users [MMN02]. Although the term similarity is often used in item-item CF, it was originally developed to recommend items to users i.e. YML recommendations [SKK01] and not MLT. Researchers often pre-generate a lists of similar items built with CF similarity to perform YML recommendations [Kar01; SKK01; CZG16]. This has been shown to be 27% better and 28× faster than the traditional user-neighbourhood based RS [Kar01]. In the CB approach, items are considered similar if they possess similar attributes [CZC15]. For movies, these usually comprise of *genre*, *director* or *cast* [PJH14; SPU02]. CB similarity could alternately be done with tags or keywords, contributed by users or domain experts; MovieLens released a set of user-contributed tags for movies via the Tag Genome Project [VSR12]. CB-similarity can also perform YML recommendations [CZG16; NK11], and improving CB similarity using supervised learning can improve YML recommendations [WAL17]. Both approaches have its own downsides; CF requires user data making it unsuitable for new items, while CB could produce only obvious recommendations. CF-CB hybrids could potentially overcome these limitations [DDV14].

Human judgement can be used to assess similarity. An absolutely correct ground truth is unlikely since the notion of similarity is subjective, but researchers aim to reach a consensus or a 'generally agreeable classification' [OD15]. Human judgement has been used to label similarity in music [JDE07; DWE05], birds [WBM10] and geometric shapes [JLC09] among others, usually to find similarity methods that match user perception. Given the massive amount of labels required for this, researchers have also investigated how to elicit confident labels at a cost-effective manner [WKB14]. We have found very few works related to movie similarity from a users perspective. In one study researchers had participants rate the similarity between 910 movie posters, but this task was for image similarity rather than movie similarity [KGA16]. In another study, researchers used low-level features in the form of subtitles to find similarity between movies [BG16].

Experiments designed to evaluate similarity in both computer and cognitive science often elicit labels in one of two ways: relative or pairwise. In relative similarity, raters are asked "is X more similar to A or B". While in pairwise similarity, raters are asked how similar is the pair X and A and then separately how similar is the pair X and B [McF12; FGM15]. Pairwise assessments often use binary labels or Likert-like scales. Researchers in music similarity found that items received more consistent labels when two levels were used ("Similar", "Not Similar"). However, the participants were more consistent with their labelling when three levels were used ("Very Similar", "Somewhat Similar", "Not Similar") [JDE07]. A binary scale is seen as less complex [DGL11] and took less time to complete [GNZ07] without compromising quality.

Research in similarity has benefited by borrowing experimental design and evaluation metrics from the Interactive Information Retrieval (IIR) community which prioritises the user in IR tasks [Kel09]. We consider IIR and item-item similarity to be analogous; in both cases the user performs a query, and receives results relevant to that topic, usually in an list ranked by estimated usefulness [Sin01]. With MLT, the query and the results are the same type of object, making it comparable to the Query-By-Example (QBE) approach [Tre00]. A comprehensive study on similarity, MLT and QBE as it relates to music can be found in [McF12], which demonstrated how elicitation of labels can improve similarity and recommendations in music.

Clough and Sanderson present a comprehensive overview of the many ways in which IR systems can be evaluated [CS13], one such method is Mean Average Precision (MAP) [SAC07]. Precision itself can be measured as either of the following [MRS08, Chapter 8]:

$$\frac{\text{(number of relevant results)}}{\text{(number of results)}} \tag{1}$$

or

$$\frac{(\#\text{true positives})}{(\#\text{true positives} + \#\text{false positives})} \qquad (2)$$

The difference in the two is in the denominator: Equation 1 includes all items returned, regardless whether or not they were labelled. We used Equation 2 which only includes items explicitly labelled true or false. Mean Average Precision first evaluates the precision of a topic (or in our case a movie), and then calculates the mean over all topics. IIR systems can be evaluated from a system perspective, which measures how well the system can rank items, or from a user perspective which measures the user satisfaction with the system [Voo01]. It has been argued that MAP is a system metric since it evaluates performance based on topics, while a more suitable measure for user satisfaction involves assessment of relevance of a fixed number of k items, such as precision@k [MRS08, Chapter 8]. However the D&M Information Systems success model introduced in 1992 [DM92] and revisited 10 years later with a survey of almost 300 journal articles [DM03] demonstrated that information quality –including relevance– leads to better user satisfaction. Thus we ourselves see MAP as a direct measurement of system performance and an indirect measurement of user satisfaction. In our research, we fix the number of items produced and evaluated per method, hence effectively measuring MAP@k which makes it a more suitable measure for user satisfaction as per recommendations above.

3 BUILDING SIMILARITY METHODS

Our goal was to compare supervised similarity methods against unsupervised methods with a user test. Here we first describe the previous work [CDL16] for context as it supplied the labels used, inspired the user interface of our study, and provided a benchmark against which we would compare our methods. Then we how we built and tested methods offline to decide which machine learning methods and features should be used in the user study, which is presented in Section 4 and Section 5.

3.1 Existing Dataset & Methods

Colucci et. al [CDL16] evaluated existing movie similarity methods from a user perspective, and showed these methods matched user perspective about half the time. They implemented four similarity methods, two based on CB similarity and two based on CF similarity. The two CF approaches were based on works by Sarwar et. al [SKK01], and used user contributed ratings of movies in MovieLens. One CF method used Pearson's correlation and another method used cosine similarity; both were implemented via the LensKit libraries from MovieLens[ELK11]. We will refer to these methods as CF-Pearson and CF-cosine respectively. The authors also implemented two CB similarity methods. The first was a blackbox from TheMovieDatabase (TMDb),[3] which used a combination of *genre* and user-contributed keyword, built with Solr's MoreLikeThis feature. The second CB approach used movie metadata from the Open Movie Database (OMDb)[4] as input, with similarity calculated using TF-IDF. Each column was weighted as follows: *title* 0.25, *genres* 0.2, *cast* 0.2, *writer* 0.15, *director* 0.1, and *plot* 0.1. We will refer to these methods as CB-keywords and CB-metadata respectively. The authors exposed a web-based front-end where participants could label the movies "similar," "not similar," or to skip if they didn't know. The results of each methods are shown in Figure 2. CB-keywords was the clear winner in their research, although we note that the authors used pairwise precision and not MAP.

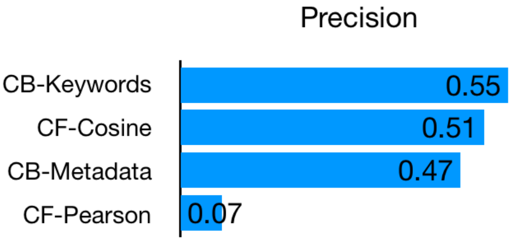


Figure 2: Perceived similarity of existing methods reported by [CDL16].

The primary purpose of their research was to evaluate similarity methods, but a byproduct was labels indicating perceived similarity. There were specifically 3803 binary labels from 14 graduate students, which we would later use to train our methods. We took a few cues from this work, with a goal of improving on it. First we used the labels collected to build and evaluate supervised machine learning models, which we hypothesised will perform better than their unsupervised methods. Second, we reused CB-keywords as-is as the benchmark in our study with the goal to outperform it. Third, we built the web front-end shown in Figure 3 to mimic the previous study.

3.2 Our Methods

Here we describe how we built and evaluated our supervised learning methods offline, with the goal of selecting the best ones for inclusion in the user study. Learning-to-Rank methods were used to produce a list of similar items where more relevant items are higher on the list [QLX10]. We tested permutations of methods described below, and selected the best methods and features combinations for the user study. All evaluation was done with leave-one-out cross validation, where

[3] https://www.themoviedb.org

[4] https://omdbapi.com

one movie (not one label) was left out, since the goal was to optimise MAP and not pairwise precision. Since there were 143 movies in the dataset, the results presented below are those averaged over 143 iterations.

We wanted to focus on CB similarity, but also aimed to build a hybrid model where similarity is predicted based on a combination of CB and CF similarity. Like Colucci et. al, we used metadata from OMDb and started with the same features: *title*, *genre*, *cast*, *writer*, *director*, and *plot*. We then added these features also from OMDb: *awards*, *country*, *full plot*, and *language*. The rationale is that two movies could be considered more similar because they were both in Mandarin, both from France, or both won the Independent Spirit Award. The *full plot* was longer than *plot* and therefore could have more relevant keywords. Previous work [CDL16] proposed that an older candidate movie may be seen as less similar than a newer one, so we engineered the feature *age difference*. Let M_1 and M_2 be the two movies for which we are evaluating similarity, then:

$$\text{Age Diff} = 1 - \frac{|\text{releaseYear}(M_1) - \text{releaseYear}(M_2)|}{\text{max(ageDiff)}} \tag{3}$$

max(ageDiff) refers to the difference between the latest M_1 and the oldest M_2 in the entire database.

We know that movie pairs with high CF similarity can be perceived to be similar [CDL16]. We further believed that CF and CB can be hybridised to produce a single method that considers both, where two movies are considered similar if they shared metadata *and* had common raters. This could reduce the possibility of an actually similar movie excluded by CB due to limited overlap in the metadata, or by CF because it has too few ratings (e.g. new movies). Of course movies that simultaneously suffer from both issues cannot be addressed by this hybrid approach. We considered using LensKit for CF, but there we observed one major issue we could not solve. Since CF-Pearson and CF-cosine used individual ratings of a movie as input, it did not work well when two movies have too few common raters. We believe this explains why CF-Pearson performed poorly before [CDL16]. Possible solutions such as adjusting the formulation or including a regularisation term was outside our scope. We instead shifted to using matrix factorisation (MF) methods, which adjusts for the number of ratings using latent factors. We tried two libraries which performed MF for RSs: myMediaLite (MML) [GRF11] and libMF [CYY16].

Three approaches were considered to calculate similarity between each features (e.g. *genre*): TF-IDF, BM25F and Jaccard similarity. While TF-IDF is a common approach, BM25F is a reasonable alternative. Three different supervised learning methods were considered for supervised learning methods: linear regression, logistic regression, and SVM with linear kernel. We wanted to try all candidate methods with CF included as a feature and without it. Since there were two CF similarity libraries to consider, we had in fact three factors: none, libMF and MML.

3.3 Method Selection

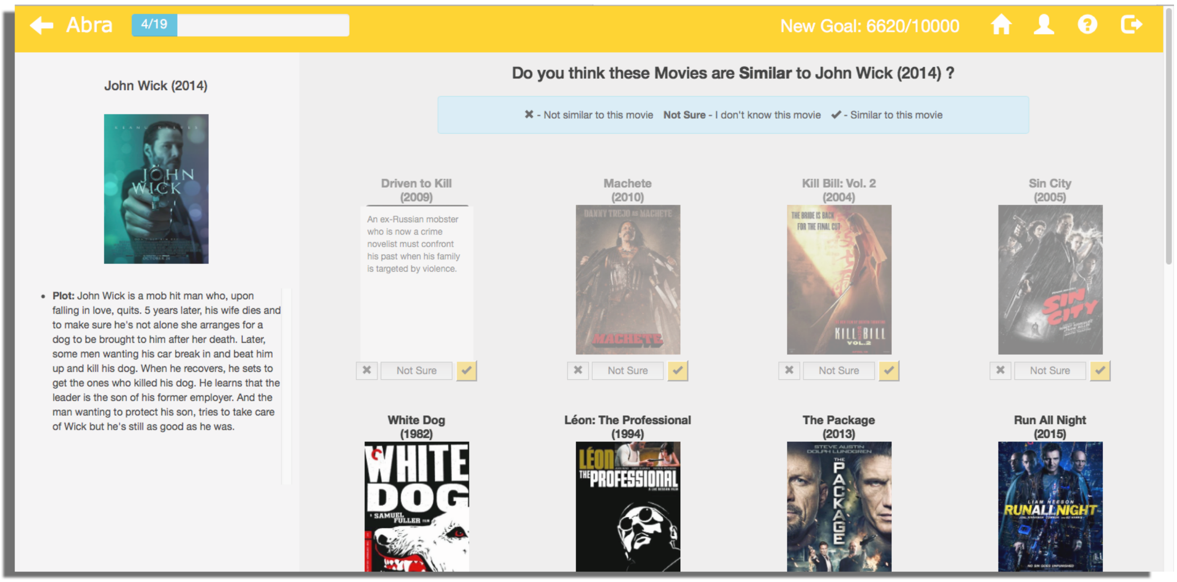
Now we move on to selecting the best methods among those described in Section 3.2 to be included in the user testing phase of our work. Note that we have considered three similarity approaches (TF-IDF, BM25F, Jaccard), three supervised learning techniques (linear, logistic, SVM) and three ways to build CF similarity (none, libMF, MML). Trying all methods would have required us to evaluate $(3 \times 3 \times 3 = 27)$ methods offline, which was too computationally expensive. So we first aimed to eliminate the least performing methods.

Between the three similarity approaches, TF-IDF provided the highest MAP on average at 0.73 followed by BM25F at 0.72 and Jaccard at 0.69. In addition to having the lowest MAP, Jaccard was also unusually slow, so it was eliminated. The three supervised methods were virtually indistinguishable; linear regression had an average MAP of 0.73, logistic regression 0.74 and SVM 0.73. We decided to only use linear regression as it was easier to explain and closer in implementation to previous work. We now had two similarity measures: BM25F and TF-IDF, and one supervised learning method: linear regression. This brought it down to a more manageable $(2 \times 1 \times 3 = 6)$ methods.

We found that the top three combinations by MAP were TF-IDF + no CF (0.71), BM25F + no CF (0.71) and BM25F+MML (0.70), and decided that these would be included in the user testing. There was little difference noticed when CF was included as a feature or not. But there were differences in compute time for different MF libraries: libMF took 1.3 minutes while MML took 11.4 minutes on average per iteration. We decided to include BM25F+libMF (0.65) in the user testing because we believed libMF had its benefits. Firstly libMF was almost 10× faster than MML; second BM25F+libMF produced a very different list than other methods, with a Jaccard Difference to BM25F of 0.48. For context BM25F+MML had a Jaccard Difference of 0.07 to BM25F. So, while BM25F was slightly less precise, it did provide quite a different list. Since diversity is known to improve user experience with YML [ZMK05; VBK14], we thought perhaps this diversity by libMF could benefit MLT too.

4 USER EXPERIMENT DESIGN

The goal of our user testing was to validate if our supervised methods could lead to better perceived similarity. The four methods chosen, as discussed in Section 3, are TF-IDF, BM25F, BM25F+MML, and BML25F+libMF.


Abra 4/19 New Goal: 6620/10000

John Wick (2014)

- **Plot:** John Wick is a mob hit man who, upon falling in love, quits. 5 years later, his wife dies and to make sure he's not alone she arranges for a dog to be brought to him after her death. Later, some men wanting his car break in and beat him up and kill his dog. When he recovers, he sets to get the ones who killed his dog. He learns that the leader is the son of his former employer. And the man wanting to protect his son, tries to take care of Wick but he's still as good as he was.

Do you think these Movies are Similar to John Wick (2014) ?

✖ - Not similar to this movie **Not Sure** - I don't know this movie ✔ - Similar to this movie

Driven to Kill (2009)

An ex-Russian mobster who is now a crime novelist must confront his past when his family is targeted by violence.

✖ Not Sure ✔

Machete (2010)

✖ Not Sure ✔

Kill Bill: Vol. 2 (2004)

✖ Not Sure ✔

Sin City (2005)

✖ Not Sure ✔

White Dog (1982)

Léon: The Professional (1994)

The Package (2013)

Run All Night (2015)

Figure 3: Experiment's user interface as per Section 4. To the left is the poster and short plot of the selected movie. To the right are eight suggested similar movies and hovering over a poster will surface a short plot. Users submit relevance feedback by indicating if the suggested movie is similar to the movie to the left. Labelled movies are greyed out. In this example there are 19 suggested similar movies, and four have been labelled.

We also included the best method from previous research as a benchmark [CDL16] (See section 3.1 and Figure 2). Our user experiment therefore simultaneously assessed five similarity methods. We built a publicly accessible website where anyone can sign up and submit annotations. Like the previous work [CDL16] the database of movies contained about 27000 movies from MovieLens with metadata from OMDb. We collected no personally identifiable information, except for some optional demographic information including work, age, and gender to evaluate diversity. We spread news of the website via social media to elicit volunteers.

After users signed up and completed demographic information, they were taken to the landing page. A randomly generated list of movies were shown as a suggestion. The user's ID was used as a seed to the randomiser to ensure the list does not look random at every refresh. A Bayesian belief network was used to ensure movies shown were representative based on genre, popularity and age. A search bar allowed users to find any movie by title. The page also showed a "goal" indicating the number of labels we wanted, and the number currently available. The goal was set to 5000, and increased in increments of 5000 when each stage was 80% reached.

Upon selecting a movie to evaluate, a user was taken to the annotation page shown in Figure 3. Six similar movies were chosen per method and merged into a list to remove duplicates. This merged list was randomised before being shown to users. Note that in the extreme case where all methods produced the same list, only six movies would be shown. In the other extreme case where all methods produced a different list, 30 movies would be shown. The number six was selected as it produced a total of 20 similar movies in the merged list on average, which is in line with the number of similar movies surfaced in previous work [CDL16].

Users were requested to supply labels by indicating if the suggested movies were similar. They could indicate the movie was similar, not similar or "not sure". Users were able to undo any actions or change any labels at any time. Candidate similar movies displayed the title and year, the plots were available on-demand. Plots were shown when they hovered over the poster with a mouse or touched the poster on a touchscreen.

5 USER EXPERIMENT RESULTS

Here we discuss the results of the user testing according to the design in Section 4. We start by describing the responses, then we evaluate the performance of similarity methods, and finally present an analysis.

5.1 Responses

A total of 136 people signed up and 114 people participated in the survey indicating a drop-out rate of 16%. Exactly 100 participants reported age, both the median and mode was 24. 72 reported their gender as male (63%), 30 as female (26%), 1 reported "others" and 11 chose to not report gender. The participants had a high education rate, with 69 in or completed a graduate program, 25 bachelors, 9 high school, and 11 not reported. In terms of employment, 65 self-reported as students while 29 were employed, 7 unemployed, 1 each reported "retired," "self-employed," or "homemaker."

	Full			In training set			Not in training set		
Method	**Mean**	**SD**	**Median**	**Mean**	**SD**	**Median**	**Mean**	**SD**	**Median**
TF-IDF	0.70	0.32	0.80	0.75	0.27	0.80	0.69	0.33	0.80
BM25F	0.69	0.32	0.78	0.72	0.30	0.79	0.68	0.33	0.76
BM25F + MML	0.70	0.32	0.80	0.73	0.30	0.82	0.70	0.32	0.80
BM25F + libML	0.65	0.36	0.71	0.75	0.26	0.76	0.62	0.36	0.67
Benchmark	0.48	0.35	0.50	0.47	0.34	0.40	0.49	0.36	0.50

Table 1: Mean Average Precision (MAP) along with standard deviation (SD) and median of each method. The first group shows the full results, the second group represents those movies that were strictly in the training set while the final group excludes all movies that were in the training set.

Participants submitting a total of 9511 responses for 393 movies, of which 6605 were binary (yes/no) while 2906 were 'not sure'. 310 of the movies were not in the training set, allowing us better claims of generalisability. On average, each user contributed a mean of 83 labels. The search function was used a total of 128 times by 28 users. There were 1087 movie pairs with binary responses from more than one user. We checked for agreement and found that 702 or 65% had complete agreement, i.e. everyone who labelled these pairs agreed that the pair was similar (or dissimilar). while 848 or 78% had at least a $2/3$ agreement.

We analysed our responses to see if it is representative of the MovieLens dataset in terms of *genre* and visualised in Figure 4. From this image we see that the movies labelled in our study is at least more representative than Colucci et. al; the top two *genre*s are almost identical. A Pearson's correlation with *genre* percentage as input showed that our dataset had $r = .96$ against all movies from MovieLens, and $r = .90$ against movies from the past 5 years of MovieLens. In contrast Colucci et. al had a correlation of $r = .49$ and $r = .55$ respectively. We consider this to mean our study is more representative in terms of *genre*.

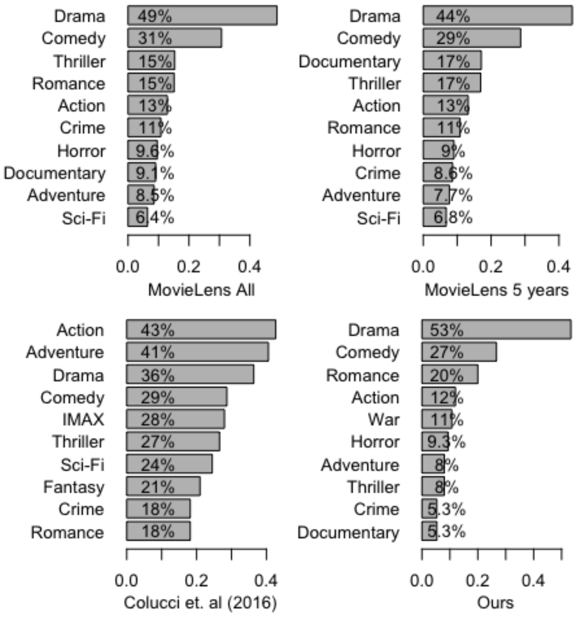


Figure 4: Top: Percentage of movies containing these *genre*s in the entire MovieLens library (L) and movies from the last five years of MovieLens (R). Bottom: Distribution for Colucci et. al [CDL16] (L), and ours (R).

5.2 Performance

We analysed the results by three groups: the first contained all movies, the second group were only where the selected movies were part of the training, while the third group were those where the selected movies were not in the training set. The last group was most important as it indicated generalisability. We used a Kruskal-Wallis test for statistical significance as our experiment used ordinal responses. This test is based on median, which we therefore report alongside the means.

We see in Table 1 that all four methods introduced in this paper outperformed the benchmark in all three groups. We performed a Kruskal-Wallis test to check for statistical significance and found that the difference was statistically significant for all three groups full ($\chi^2 = 86.868, df = 4, p < .001$), in training set ($\chi^2 = 35.429, df = 4, p < .001$) and not in training set ($\chi^2 = 58.016, df = 4, p < .001$). Hence we ran a post-hoc test with Dunn's t-test and Holm-Bonferroni correction. All pairwise evaluations involving the benchmark were statistically significant ($p < .05$) while those that did not involve benchmark were not. Practical significance between the benchmark and our methods in Cohen's d, is in Table 2. A common interpretation of Cohen's d is that .2 means the practical significance is small, .5 is medium but visible to the naked eye, and .8 is large [SF12].

	Full	In train.	Not in train.
TF-IDF	0.63	0.90	0.57
BM25F	0.61	0.79	0.57
BM25F + MML	0.65	0.81	0.61
BM25F + libML	0.47	0.92	0.36

Table 2: Effect size in Cohen's d against the benchmark. Grouped by all movies, movies strictly in the training set, and movies strictly not in the training set.

5.3 Analysis & Implication

The main findings is that the use of labels to train a supervised model results in an improvement in perceived similarity. It is also noted that there are differences when we account for movies that were in the training set, the precision dropped a little. This drop was most apparent for BM25F+libML.

An interesting finding to us was that there were no statistical significance between different models. The biggest difference was between BM25F+libML vs. BM25F+MML for items not in the training set, with a difference in MAP of 0.08. It had a Dunn's post-hoc test of $p = .2016$ and effect size measured with Cohen's d of .22. We believe that this comparison could be statistically significant with a small effect size in a live deployment or another experiment with larger number of movies and participants

Movies that were in the training set appear to have higher precision and performed better against the benchmark in Cohen's d. This is unsurprising but highlights the importance of evaluating such methods based on items not in the training set to infer generalisability.

6 DISCUSSION

We believe there can be reasonable confidence in the experiment presented here. There was a high number of movies in our evaluation that were not in the training set, which speaks to the generalisability of the methods. The inter-rater agreement of 65% for complete agreement indicated reliability of participants. The distribution of movies by *genre* were also representative of movies in the library.

The fact that BM25F+libML had lower MAP than other supervised methods was unsurprising considering it had slightly lower performance during the machine-learning stage. This difference was not statistically significant but it could be in a setting with larger participants and/or movies. We should also point out that there are two benefits to this method: (1) it is faster than MML, and (2) it produced similar movies notably different from all other supervised methods. It could be more suitable for use in practice when timeliness and resource consumption matters, or by researchers eager to test out different variations quickly. In addition, it could be used in cohorts with other similarity methods to produce a more diverse list of similar items.

We were admittedly surprised that BM25F showed no improvement over TF-IDF. In fact both TF-IDF and BM25F had the exact effect size over the benchmark measured in Cohen's d. There was also no noticeable difference in the time it took for both to complete. The inclusion of collaborative similarity as a feature in the hybrid method BM25F+MML seems to have some improvements noted in the effect size for movies not in the training set. But this was not statistically significant and admittedly lower than we expected. Future researchers could investigate this further, including to identify which types of movie benefit from the hybrid approach. For now, our recommendation to researchers using our generated list of similar item should use BM25F+MML, while researchers who wish to build from scratch using our similarity labels only could start by implementing TF-IDF.

This work opens up a number of research question which we encourage future researchers to explore, we list a few such questions here. We believe more work needs to be done to understand *why* people believe two movies are similar. This could be used to build better machine learning methods including personalised similarity methods and to provide a better experience overall.s It may be possible to analyse our data to identify which features are most salient in predicting similarity, and likewise if different people have different preferences. We believe a similar study could be done to improve similarity in TV series, songs, or video games.

It is evident here that labels are useful in the prediction of similarity. While this is not surprising, this is the first time it has been shown to be true in finding similar movies. Future work should include elicitation of labels from many more subjects and for many more movies to ensure higher coverage and better confidence. We therefore release all labels collected during this study, source code for the website to elicit labels, and all intermediate data generated during the process in order to encourage other researchers to build on our work. Our dataset is a notable improvement over Colucci et. al [CDL16] in terms of number of labels, number of participants, and genre representation. We hope this would lead to even better machine learning models, which will improve user experience by helping them find similar movies.

7 CONCLUSION

In this paper we showed that finding similar movies can be improved if we use human-annotated data representing perceived similarity in movies. We tested a few machine learning options offline, identified the best methods and features, and then evaluated with users. Our methods demonstrated significant improvement over the benchmark introduced in previous work which was built with unsupervised machine learning. The four supervised methods in our user testing were not statistically significant between themselves, which indicated that in a small sample such as ours any of our methods could produce the same experience. We showed that there is more that can and should be done to improve user experience with similar items. We release our dataset to encourage and enable future research in this domain by both HCI and ML researchers.

REFERENCES

[BG16] Bougiatiotis, K. and Giannakopoulos, T. Content Representation and Similarity of Movies based on Topic Extraction from Subtitles. Proceedings of the 9th Hellenic Conference on Artificial Intelligence. ACM. 2016,

[BHY97] Burke, R. D., Hammond, K. J., and Yound, B. The FindMe approach to assisted browsing. IEEE Expert 12.4 (1997),

[BL07] Bennett, J. and Lanning, S. The netflix prize. Proceedings of KDD cup and workshop. Vol. 2007. 2007,

[CDL16] Colucci, L., Doshi, P., Lee, K.-L., Liang, J., Lin, Y., Vashishtha, I., Zhang, J., and Jude, A. Evaluating Item-Item Similarity Algorithms for Movies. Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems. ACM. 2016,

[CS13] Clough, P. and Sanderson, M. Evaluating the performance of information retrieval systems using test collections. Information Research 18.2 (2013).

[CYY16] Chin, W.-S., Yuan, B.-W., Yang, M.-Y., Zhuang, Y., Juan, Y.-C., and Lin, C.-J. LIBMF: a library for parallel matrix factorization in shared-memory systems. The Journal of Machine Learning Research 17.1 (2016),

[CZC15] Chang, S., Zhou, J., Chubak, P., Hu, J., and Huang, T. S. A space alignment method for cold-start TV show recommendations. Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI. 2015,

[CZG16] Chen, Y., Zhao, X., Gan, J., Ren, J., and Hu, Y. Content-based top-n recommendation using heterogeneous relations. Australasian Database Conference. Springer. 2016,

[DDV14] Dooms, S., De Pessemier, T., Verslype, D., Nelis, J., De Meulenaere, J., Van den Broeck, W., Martens, L., and Develder, C. OMUS: an optimized multimedia service for the home environment. Multimedia tools and applications 72.1 (2014),

[DGL11] Dolnicar, S., Grün, B., and Leisch, F. Quick, simple and reliable: Forced binary survey questions. International Journal of Market Research 53.2 (2011),

[DM03] Delone, W. H. and McLean, E. R. The DeLone and McLean model of information systems success: a ten-year update. Journal of management information systems 19.4 (2003),

[DM92] DeLone, W. H. and McLean, E. R. Information systems success: The quest for the dependent variable. Information systems research 3.1 (1992),

[DWE05] Downie, J., West, K., Ehmann, A., and Vincent, E. The 2005 music information retrieval evaluation exchange (mirex 2005): Preliminary overview. 6th Int. Conf. on Music Information Retrieval (ISMIR). 2005,

[EGG12] Estes, Z., Gibbert, M., Guest, D., and Mazursky, D. A dual-process model of brand extension: taxonomic feature-based and thematic relation-based similarity independently drive brand extension evaluation. Journal of Consumer Psychology 22.1 (2012),

[ELK11] Ekstrand, M. D., Ludwig, M., Konstan, J. A., and Riedl, J. T. Rethinking the Recommender Research Ecosystem: Reproducibility, Openness, and LensKit. Proceedings of the Fifth ACM Conference on Recommender Systems. RecSys '11. Chicago, Illinois, USA: ACM, 2011,

[FGM15] Fisher, A. V., Godwin, K. E., Matlen, B. J., and Unger, L. Development of Category-Based Induction and Semantic Knowledge. Child development 86.1 (2015),

[GNZ07] Grassi, M., Nucera, A., Zanolin, E., Omenaas, E., Anto, J. M., and Leynaert, B. Performance Comparison of Likert and Binary Formats of SF-36 Version 1.6 Across ECRHS II Adults Populations. Value in Health 10.6 (2007),

[GRF11] Gantner, Z., Rendle, S., Freudenthaler, C., and Schmidt-Thieme, L. MyMediaLite: a free recommender system library. Proceedings of the fifth ACM conference on Recommender systems. ACM. 2011,

[HK16] Harper, F. M. and Konstan, J. A. The movielens datasets: History and context. ACM Transactions on Interactive Intelligent Systems (TiiS) 5.4 (2016),

[JDE07] Jones, M. C., Downie, J. S., and Ehmann, A. F. Human Similarity Judgments: Implications for the Design of Formal Evaluations. ISMIR. 2007,

[JLC09] Jagadeesan, A. P., Lynn, A., Corney, J. R., Yan, X., Wenzel, J., Sherlock, A., and Regli, W. Geometric reasoning via internet crowdsourcing. 2009 SIAM/ACM Joint Conference on Geometric and Physical Modeling. ACM. 2009,

[JW02] Jeh, G. and Widom, J. SimRank: a measure of structural-context similarity. Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining. ACM. 2002,

[Kar01] Karypis, G. Evaluation of item-based top-n recommendation algorithms. Proceedings of the tenth international conference on Information and knowledge management. ACM. 2001, [Kel09] Kelly, D. Methods for evaluating interactive information retrieval systems with users. Foundations and Trends in Information Retrieval 3 (2009),

[KGA16] Kleiman, Y., Goldberg, G., Amsterdamer, Y., and Cohen-Or, D. Toward semantic image similarity from crowdsourced clustering. The Visual Computer 32.6-8 (2016),

[McF12] McFee, B. More like this: machine learning approaches to music similarity. PhD thesis. University of California, San Diego, 2012.

[MG12] Mirman, D. and Graziano, K. M. Individual differences in the strength of taxonomic versus thematic relations. Journal of experimental psychology: General 141.4 (2012),

[MMN02] Melville, P., Mooney, R. J., and Nagarajan, R. Content-boosted Collaborative Filtering for Improved Recommendations. Eighteenth National Conference on Artificial Intelligence. Edmonton, Alberta, Canada: American Association for Artificial Intelligence, 2002,

[MRS08] Manning, C. D., Raghavan, P., and Schütze, H. *Introduction to Information Retrieval*. Cambridge University Press, 2008.

[Nie08] Nielsen, J. Bridging the designer-user gap (2008).

[NK11] Ning, X. and Karypis, G. Slim: Sparse linear methods for top-n recommender systems. Data Mining (ICDM), 2011 IEEE 11th International Conference on. IEEE. 2011,

[OD15] Organisciak, P. and Downie, J. S. Improving Consistency of Crowdsourced Multimedia Similarity for Evaluation. Proceedings of the 15th ACM/IEEE-CS Joint Conference on Digital Libraries. ACM. 2015,

[PJH14] Pirasteh, P., Jung, J. J., and Hwang, D. Item-based collaborative filtering with attribute correlation: a case study on movie recommendation. Intelligent Information and Database Systems. Springer, 2014,

[QLX10] Qin, T., Liu, T.-Y., Xu, J., and Li, H. LETOR: A benchmark collection for research on learning to rank for information retrieval. Information Retrieval 13.4 (2010),

[SAC07] Smucker, M. D., Allan, J., and Carterette, B. A comparison of statistical significance tests for information retrieval evaluation. Proceedings of the sixteenth ACM conference on Conference on information and knowledge management. ACM. 2007,

[SF12] Sullivan, G. M. and Feinn, R. Using effect size-or why the P value is not enough. Journal of graduate medical education 4.3 (2012),

[Sin01] Singhal, A. Modern information retrieval: A brief overview. IEEE Data Eng. Bull. 24.4 (2001),

[SKK01] Sarwar, B., Karypis, G., Konstan, J., and Riedl, J. Item-based Collaborative Filtering Recommendation Algorithms. Proceedings of the 10th International Conference on World Wide Web. WWW '01. Hong Kong, Hong Kong: ACM, 2001,

[SPU02] Schein, A. I., Popescul, A., Ungar, L. H., and Pennock, D. M. Methods and metrics for cold-start recommendations. Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval. ACM. 2002,

[Tre00] Trewin, S. Knowledge-based recommender systems. Encyclopedia of library and information science 69.Supplement 32 (2000),

[VBK14] Vargas, S., Baltrunas, L., Karatzoglou, A., and Castells, P. Coverage, Redundancy and Size-awareness in Genre Diversity for Recommender Systems. Proceedings of the 8th ACM Conference on Recommender Systems. RecSys '14. Foster City, Silicon Valley, California, USA: ACM, 2014,

[Voo01] Voorhees, E. M. The philosophy of information retrieval evaluation. Workshop of the Cross-Language Evaluation Forum for European Languages. Springer. 2001,

[VSR12] Vig, J., Sen, S., and Riedl, J. The tag genome: Encoding community knowledge to support novel interaction. ACM Transactions on Interactive Intelligent Systems (TiiS) 2.3 (2012),

[WAL17] Wang, C., Agrawal, A., Li, X., Makkad, T., Veljee, E., Mengshoel, O., and Jude, A. Content-Based Top-N Recommendations With Perceived Similarity. IEEE International Conference on Systems, Man, and Cybernetics (SMC). 2017.

[WB99] Wisniewski, E. J. and Bassok, M. What makes a man similar to a tie? Stimulus compatibility with comparison and integration. Cognitive Psychology 39.3 (1999),

[WBM10] Welinder, P., Branson, S., Mita, T., Wah, C., Schroff, F., Belongie, S., and Perona, P. Caltech-UCSD birds 200 (2010).

[WKB14] Wilber, M. J., Kwak, I. S., and Belongie, S. J. Cost-effective hits for relative similarity comparisons. Second AAAI Conference on Human Computation and Crowdsourcing. 2014.

[ZMK05] Ziegler, C.-N., McNee, S. M., Konstan, J. A., and Lausen, G. Improving Recommendation Lists Through Topic Diversification. Proceedings of the 14th International Conference on World Wide Web. WWW '05. Chiba, Japan: ACM, 2005,