Navigating the Tag Genome

Jesse Vig
Department of Computer Science and Engineering
University of Minnesota
jvig@cs.umn.edu

Shilad Sen
Math, Statistics, and Computer Science Department
Macalester College
ssen@macalester.edu

John Riedl
Department of Computer Science and Engineering
University of Minnesota
riedl@cs.umn.edu

ABSTRACT

Tags help users understand a rich information space, by showing them specific text annotations for each item in the space and enabling them to search by these annotations. Often, however, users may wish to move from one item to other items that are similar overall, but that differ in key characteristics. For example, a user who loves Pulp Fiction might want to see a similar movie, but might be in a mood for a less "dark" movie. This paper introduces Movie Tuner, a novel interface that supports navigation from one item to nearby items along dimensions represented by tags. Movie Tuner is based on a data structure called the tag genome, which is described in separate work. The tag genome encodes each item's relationship to a common set of tags by applying machine learning algorithms to user-contributed content. The present paper discusses our design of Movie Tuner, including algorithms for navigating to new items and for suggesting tags for navigation. We present the results of a 7-week field trial of 2,531 users of Movie Tuner, and of a survey evaluating users' subjective experience.

Author Keywords

tagging, recommender systems, conversational recommenders

ACM Classification Keywords

H.5.3 Information Interfaces and Presentation: Group and Organization Interfaces—*Collaborative computing*; H.5.2 Information Interfaces and Presentation: User Interfaces

General Terms

Design, Experimentation, Human Factors

1. INTRODUCTION

Tagging systems have become increasingly popular on the Web. Users of tagging systems create free-form text descriptors of music, pictures or encyclopedia articles and use these descriptors to navigate complex information spaces. However, tags present challenges when used in navigation. Tagging systems lack the hierarchical structure of expert-designed taxonomies like the Dewey Decimal System [15]. Users searching for an item must specify a tag capturing their query instead of drilling down through system-specified alternatives. Studies have shown that some users find it difficult to think of tags [13].

In this paper we explore a novel form of navigation that is based on tags, but that offers a fundamentally different form of navigation than traditional tagging systems. We motivate our system with a hypothetical dialogue between a movie navigation system and a user Marco:

Marco: *I'd like to watch a movie, but I'm not exactly sure what I want.*
System: *How about When Harry Met Sally, Up, or Reservoir Dogs?*
Marco: *Reservoir Dogs looks like a possibility, please tell me more.*
System: *It is a classic, nonlinear, violent, crime, cult film.*
Marco: *I'm not in the mood for something quite that violent.*
System: *Then how about The Usual Suspects? It's like Reservoir Dogs, but less violent.*
Marco: *I'll take it!*

Movie Tuner is a novel application that enables users to navigate an information space much like Marco did. Figure 1 shows the Movie Tuner interface as Marco might have seen it after selecting *Reservoir Dogs*. The interface displays a set of tags (*violent*, *crime*, *nonlinear*, *classic*, *cult film*) each with a relevance meter indicating how strongly *Reservoir Dogs* exhibits that quality. Marco clicked the *less* button alongside the *violent* tag; in response, the system displayed a list of movies that are "Similar to Reservoir Dogs, but less violent", including *The Usual Suspects*, which Marco eventually chose.

Navigating a space by critiquing specific items, known as *example critiquing*, has been studied in the context of knowledge-based systems. For example, in the Entree system users critique restaurant recommendations based on price ("less expensive"), style ("more traditional"), atmosphere ("quieter"), and other criteria [1]. In Movie Tuner, users critique items with respect to tags. Three advantages of tag-based critiquing versus traditional example critiquing are: (1) the tags are dimensions chosen by users, and hence are expressed in their own language; (2) tags are resources

IUI 2011, February 13 - 16, 2011, Palo Alto, California, USA.

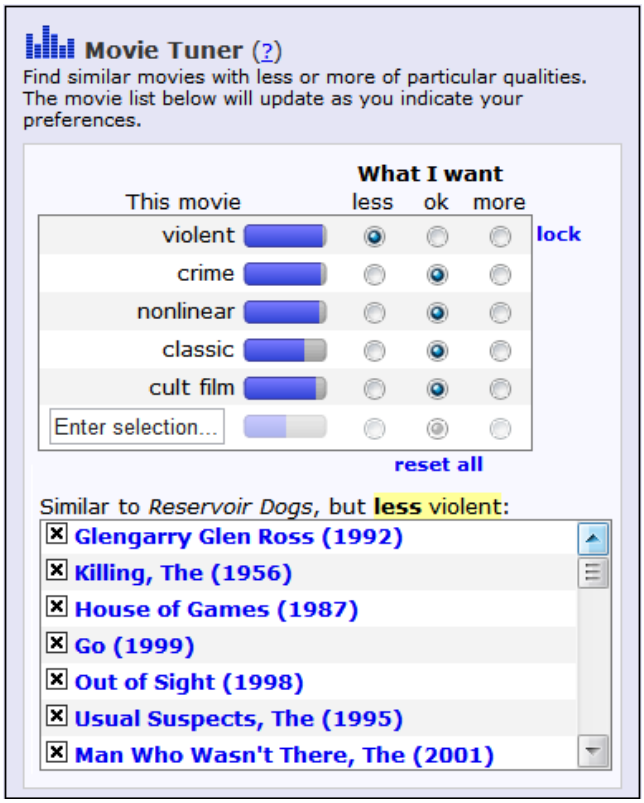

Figure 1: Movie Tuner interface for *Reservoir Dogs*, after a user applies the critique "less violent".

that are freely generated by users of the system, with only modest support from the system designers [15]; and (3) tags describe both factual and subjective aspects of items [13]. The design of Movie Tuner focuses on two primary interactions from example-critiquing systems: *applying critiques* refers to how users tell the system what they would like to change about an item ("I'd like something less violent than Reservoir Dogs"), and *responding to critiques* describes how the system chooses new items in response to a user's critiques ("Here are the movies similar to Reservoir Dogs, but less violent...").

This form of navigation has several key differences versus traditional tag-based navigation. First, tags are used to react to specific items ("I'd like something less violent than *Reservoir Dogs*"), in contrast to traditional tag search, where users begin by specifying one or more tags ("I'd like something *not violent*"). Presenting tags in the context of specific items of interest to users is consistent with prior studies that suggest that people formulate preferences by interacting with the available choices rather than deciding in advance what they want [10]. Second, the system explicitly models the relevance of tags to items on a consistent 0-1 scale. This allows the system to compare items with respect to tags ("The Usual Suspects is *less violent* than Reservoir Dogs").

A challenge in tag-based critiquing is developing algorithms that extract a structured knowledge base from unstructured user tagging activities. Movie Tuner is driven by an underlying data structure called the *tag genome*, described in [18], that is built automatically by applying machine-learning to user-contributed content. The tag genome provides the data used by Movie Tuner to display the relevance of tags to items, to compare items with respect to particular tags, and to find items that are similar to a given item.

In this paper, we first summarize related work and discuss how Movie Tuner builds on the existing critiquing and tagging literatures. We then provide an overview of the tag genome and the MovieLens research platform. Next, we detail the design of applying critiques and responding to critiques in Movie Tuner. We then present the results of a 7-week field study of Movie Tuner.

2. RELATED WORK

We describe two types of navigation systems that build on traditional tagging and recommender systems. Tag-based recommenders combine features of tagging and rating systems, while example-critiquing systems enhance traditional recommenders by enabling rich feedback.

2.1 Tag-based recommenders

Recent work has explored systems that combine tagging and recommendation [14, 9, 5]. The most similar work to Movie Tuner is the Music Explaura system [5], in which users "steer" music recommendations using tags. MrTaggy [6] is a tagging system that supports exploration by enabling users to provide positive or negative feedback to tags associated with particular items.

Movie Tuner differs from these systems in several ways. First, Movie Tuner provide an explicit measure of tag relevance on a 0-1 scale that is based on a gold-standard set of tag relevance values provided by users. Second, Movie Tuner provides a novel interface for visualizing tag relevance and applying critiques. Third, we evaluate Movie Tuner in a live user study involving thousands of users, comparing multiple algorithms for suggesting tags as well as multiple algorithms for retrieving items in response to users' critiques.

2.2 Example-critiquing systems

Researchers have explored conversational recommenders that allow users to give immediate feedback on recommendations and then adjust recommendations accordingly [7, 1, 3, 17]. One type of feedback supported by these systems is a *critique*, which describes what the user thinks is wrong with a particular example. For example, in the QwikShop system for digital cameras, users may apply critiques such as "less expensive", "more memory", or "higher resolution" [8]. The system then responds by selecting a new set of items that satisfies the user's critique. This type of conversational recommender is often referred to as an *example-critiquing* system.

Example-critiquing systems generally offer the user a narrow set of dimensions for critiquing items, and these dimensions are typically chosen by designers of the system. For example, in the QwikShop system mentioned above, cri-

tique dimensions include manufacturer, zoom level, memory, weight, resolution, size, case, and price. Moreover, example-critiquing systems are traditionally knowledge-based; for example, the Entree recommender system has an underlying database with the cuisine, price, style, and atmosphere of every restaurant in the system.

The example-critiquing paradigm motivates our design of Movie Tuner. In Movie Tuner, tags serve as the dimensions along which users critique items. For example, users may ask for a movie that is "more funny" or "less violent" because *funny* and *violent* are tags in the system. However, in contrast to the compact set of system-engineered dimensions typically provided by example-critiquing systems, tags provide a broad range of feedback in the language of the users themselves. Further, Movie Tuner requires no underlying knowledge base that knows, for example, how violent *Die Hard* is, or how much action is in *Forrest Gump*. Rather, this information is generated automatically by machine learning models based on user-contributed content [18].

3. FRAMEWORK

3.1 The tag genome

Just as an organism is defined by a sequence of genes, an item in an information space may be defined by its relationship to a set of tags [18]. If T is a set of tags and I is a set of items, we quantify the relationship between each item $i \in I$ and tag $t \in T$ by the *relevance* of t to i, denoted as $\mathrm{rel}(i,t)$. $\mathrm{rel}(i,t)$ measures how strongly tag t applies to item i on a continuous scale from 0 (does not apply at all) to 1 (applies very strongly). In the movie domain, for example, rel(*Reservoir Dogs*, *violent*) = 0.98, rel(*The Usual Suspects*, *violent*) = 0.65, and rel(*A Cinderella Story*, *violent*) = 0.03.

The *tag genome* for an item i is the vector of tag relevance values across all tags in T, denoted as $\mathrm{rel}(i)$. Formally,

$$\mathrm{rel}(i) = \big\langle \mathrm{rel}(i,t_1), \ldots, \mathrm{rel}(i,t_n) \big\rangle \; \forall t_k \in T$$

In separate work we showed how to construct the tag genome by applying machine learning algorithms to user-contributed content [18]. Specifically, we constructed a hierarchical regression model that predicts the relevance of an arbitrary (*item*, *tag*) pair using features extracted from tags, ratings, and text reviews. We trained the model with a gold standard of 50,203 (*item*, *tag*) relevance values provided by users.

The tag genome has three key features that support example-critiquing. First, the tag genome provides a continuous measure of tag relevance on a consistent 0-1 scale. Second, the tag genome is dense, in that it defines a relevance value for every tag $t \in T$, enabling comparisons between items with respect to arbitrary tags. Third, the tag genome may be used to measure similarity between items so the system can find similar items when responding to critiques.

3.2 The MovieLens platform

We used the MovieLens[1] movie recommender system as a platform for implementing Movie Tuner. The primary purpose of MovieLens is movie recommendation: users rate movies on a scale of 1 to 5 stars and receive recommendations in return. MovieLens has been in continuous use since 1997, and 186,000 users have provided a total of 17 million movie ratings. MovieLens also supports tagging of movies; 5,375 users have applied 31,325 distinct tags, resulting in over 246,000 total tag applications.

We added the Movie Tuner interface to two screens on MovieLens: the *movie details* page, and the *movie list* page. The movie details page displays detailed information about a particular movie including cast, director, a Netflix synopsis, a tag cloud for the movie, and Movie Tuner. The movie list page is used to display any list of movies, including search results and personalized recommendations. We added an icon users may click to see Movie Tuner. We do not show Movie Tuner for the least popular movies (< 50 ratings), because these movies tended to have too little data to accurately compute the tag genome. In total, we display Movie Tuner for 8,871 distinct movies.

4. APPLYING CRITIQUES

Users apply critiques to tell the system what they wish to change about a particular item, for example "less violent" or "more action". Below we outline the design space for how users may apply critiques, and we discuss our design decisions.

4.1 Critique dimensions

Critique dimensions represent the dimensions along which users may critique an item. In Movie Tuner, tags serve as critique dimensions. For example, some of the tags on MovieLens are *action, violent*, and *quirky*; with these tags as critique dimensions, a users might request a movie that has "more action", is "less violent", or is "more quirky".

One question is how to choose the tags that will serve as critique dimensions. On MovieLens, users have applied 31,325 distinct tags, ranging from popular tags such as *classic* (applied by 416 users), *funny* (279 users), and *animation* (236 users) to tags only applied by a single user such as *oh yah*, *sidecar*, or *acorn*. One possible design choice is to include all tags as critique dimensions. We decided instead to filter tags based on popularity and quality, retaining only those tags that we felt users would care about and would be useful in critiques. We only include tags applied by at least 10 users, since tags below this threshold tended to be either personal (*jb's dvds*), extremely specific (*archery*), or misspellings of more popular tags (*Quinten Tarantino*). We filter the remaining tags based on a tag quality metric developed in [12], excluding tags that scored in the bottom 5 percentile. After filtering, 1,570 tags remained as critique dimensions.

As shown in Figure 1, Movie Tuner displays tags in a list, with a *relevance meter* next to each tag indicating its relevance to the current item. (We discuss later how the system chooses the tags to display.) Other visualizations should work as well, such as a tag cloud with varying font size [5]. We used the relevance meter to more precisely represent the 0 to 1 relevance scale.

[1] www.movielens.org

4.2 Critique direction

In most example-critiquing systems, users critique an item by specifying a *direction* along a critique dimension, for example "*less* expensive". However, some example-critiquing systems also enable the user to provide a *magnitude*, for example "*at least $100* cheaper" [2]. Although specifying the magnitude gives users more control over their critiques, it requires more fine-grained input from the user.

In Movie Tuner, we chose to use a direction-only approach. Enabling users to specify the magnitude of the critique, perhaps with a slider, would give users additional control, but we chose the direction-only approach because it requires lower cognitive load. We denote an individual critique as a tuple (t, d), for tag $t \in T$ and direction $d \in \{-1, +1\}$, where -1 indicates *less* and $+1$ indicates *more*.

As shown in Figure 1, users choose a critique direction by clicking a "less" or "more" radio button next to a particular tag. The default "ok" selection indicates that the user does not wish to apply a critique with respect to the tag. As an alternative to the three radio buttons, we had also considered having two checkboxes, one for "less" and one for "more", but found that in initial trials users felt compelled to check one box for every tag shown. With a default selection of "ok", users understood that they could simply ignore a particular tag.

4.3 Unit versus compound critiques

A *unit critique* is constrained to a single critique dimension ("less violent"), while a *compound critique* [17, 19] spans multiple dimensions, ("less violent and more action"). Although compound critiques enable faster navigation, they also require more work from the user at each step.

Movie Tuner supports both unit and compound critiques. To apply a compound critique, users must explicitly *lock* the original critique to keep it in effect as they choose additional critiques (see Figure 1); otherwise, the original critique will be reset to the "ok" position when they select additional critiques. We require explicit locking because in initial trials users often forgot to undo their original critique before selecting other unit critiques. As a result the critiques became increasingly complex, and did not match the users' intentions.

4.4 System-suggested versus user-initiated critiques

In systems that provide a small number of critique dimensions, a common design choice is to display all critique dimensions and let users choose from them when applying critiques. In Movie Tuner, however, the number of critique dimensions (i.e. tags), far exceeds the available screen space. We considered two alternatives: in a *system-suggested* model, the system displays a small set of possible tags and users choose among them, while in a *user-initiated* model, users must enter the tags they wish to use in critiques.

We chose a mixed-initiative model where users may either choose from a set of system-suggested tags, or enter additional tags of their own. We chose to suggest tags because studies have shown some users have difficulty thinking of tags [13]. As shown in Figure 1, Movie Tuner displays 5 system-selected tags for each item. We display 5 tags per item in order to provide users with a variety of choices while conserving screen space.

Users may also enter tags not suggested by the system in an auto-complete text box ("Enter selection") below the tags currently displayed; however, users may only enter tags that are among the 1,570 tags included as critique dimensions. Once entered, the tag is displayed above along with its relevance meter as well as radio buttons for setting the critique direction. Users may also use the text box simply to inquire about the relevance of a tag to an item ("How *realistic* is The Bourne Identity?").

4.5 Tag selection algorithm

We now describe how we choose the tags to display for a particular item. We select tags based on three objectives: we choose tags that are *valuable for critiquing* an item, because the primary purpose of displaying the tags is to help users apply critiques; we choose *popular* tags, because the tags should be ones that users care about; and we choose *diverse* tags, because an orthogonal set of tags enables more efficient navigation. Below we define metrics for each of these objectives, and we describe a multi-objective optimization algorithm for selecting the tags to display.

Critique value. We define two metrics for evaluating how useful a tag is for critiquing a particular item. One metric favors *descriptive* tags; for example, *violent* is highly descriptive for *Reservoir Dogs* because it is an extremely violent movie. The other metric favors tags that *discriminate* among the space of similar items; for example, *action* is a discriminating tag for *Reservoir Dogs*, because many similar movies have either more action (e.g. *Kill Bill Vol. 1*) or less action (e.g. *Sexy Beast*).

To measure how *descriptive* a tag t is with respect to an item i, we simply use $\mathrm{rel}(i, t)$, the relevance of t to i (see Section 3.1). To measure how *discriminating* a tag t is with respect to an item i, we define a metric called *critique entropy*, which measures how evenly t separates the items neighboring i.

To compute critique entropy for tag t relative to item i, we partition the set N of neighbors of i (defined in Section 5.3) into 3 subsets N_{+1}, N_{-1}, and N_0. N_{+1} comprises neighbors of i that satisfy the critique "more t", N_{-1} comprises neighbors of i that satisfy the critique "less t", and N_0 comprises the remaining neighbors of i. Formally,

$$N_{+1} = \{j | j \in N, \mathrm{rel}(j, t) > \mathrm{rel}(i, t) + 0.25\}$$
$$N_{-1} = \{j | j \in N, \mathrm{rel}(j, t) < \mathrm{rel}(i, t) - 0.25\}$$
$$N_0 = N - N_{+1} \cup N_{-1}$$

We chose the value of 0.25 based on our qualitative analysis over a series of test cases.

This is a simplified version of the critique satisfaction model presented in Section 5, and is only used for the purpose of computing critique entropy.

We define critique entropy to be the Shannon entropy of the distribution of items over N_{+1}, N_{-1}, N_0. Formally,

$$\text{critique-entropy}(i,t) = \sum_{d \in \{+1,-1,0\}} -\frac{|N_d|}{|N|} \cdot \log\Big(\frac{|N_d|}{|N|}\Big)$$

Just as Shannon entropy measures the evenness of a distribution, critique entropy measures how evenly the critiques associated with a tag divide the space of neighboring items.

Popularity. We measure tag popularity by the number of distinct users who have applied a tag t, denoted as $\text{tag-pop}(t)$. We apply a log transform to tag-pop to make the distribution more normal.

Diversity. We measure the diversity of a set of tags based on how dissimilar the tags are to one another. To measure similarity between two tags t and u, we take the cosine similarity of their relevance values across all items in I (see Section 3.1), which we denote as $\text{tag-sim}(t,u)$. Later we will show how we use this tag similarity metric to choose a diverse set of tags.

Multi-objective optimization. Since we wish to satisfy three different objectives (critique value, popularity, diversity) simultaneously, we express the problem of choosing tags as a multi-objective optimization problem [4]. One approach for solving multi-objective optimization problems is to define an *aggregate objective function* that takes all objectives into account and computes a single utility value for each candidate solution. One may also frame the problem as a *constrained optimization problem*, where some of the objectives are expressed as constraints while others are included in the objective function.

We chose to express the tag selection problem as a constrained multi-objective optimization problem over the space of all tag sets of size 5. We define an aggregate objective function that evaluates each candidate tag set based on the objectives described above, and we also set constraints to ensure that the chosen tag set satisfies each objective to a minimal degree. We constructed two versions of the optimization problem, one that measures critique value based on tag relevance (favors descriptive tags) and one that measures critique value based on critique entropy (favors discriminating tags).

We chose the specific problem formulation below based on a series of trials with various objective functions and constraint combinations. We did not include diversity in the objective function, because we found that simply setting a constraint based on maximum pairwise similarity between tags produced sufficiently diverse tag sets. We combine popularity and critique value in the objective function by taking their product. We preferred this approach to a weighted sum because, since it is scale invariant, it requires no parameter estimation. We then add these values for all tags in the set in order to produce a single value for the entire set.

Descriptive	Discriminating
sci-fi (0.99)	fantasy (0.50)
comedy (0.98)	space (0.67)
action (0.95)	superhero (0.37)
adventure (0.85)	future (0.35)
comic book (0.75)	tense (0.38)

Table 1: Tags chosen for *Men in Black* by each tag-selection algorithm. Relevance values are shown in parentheses.

Problem formulation. Given an item i, find the set of tags $S \subset T$ that maximizes the following objective function:

$$\begin{aligned} \underset{S}{\text{maximize}} \quad & \Big\{\sum_{t \in S} \text{critique-value}(i,t) \cdot \log(\text{tag-pop}(t))\Big\} \\ \text{subject to} \quad & |S| = 5 \\ & \text{tag-pop}(t) \geq 50 \quad \forall t \in S \\ & \text{tag-sim}(t,u) < 0.5 \quad \forall (t,u) \in S,\ t \neq u \end{aligned}$$

We designed two versions of the objective function, one where $\text{critique-value}(i,t) = \text{rel}(i,t)$ (favors descriptive tags), and one where $\text{critique-value}(i,t) = \text{critique-entropy}(i,t)$ (favors discriminating tags). In the latter case, we added the following constraint[2]:

$$\text{critique-entropy}(t) \geq 0.325 \quad \forall t \in S$$

Table 1 shows an example of the tags each version generates.

Because finding exact solutions to combinatorial optimization problems is computationally expensive, we designed a greedy algorithm to find an approximate solution. The algorithm begins with an empty set of tags, then iteratively adds the tag that maximizes the objective function subject to its constraints, stopping when the size of the tag set equals 5.

5. RESPONDING TO CRITIQUES

After a user critiques an item, the system must respond by retrieving new items that satisfy the critique. In this section we describe the algorithm for responding to critiques on Movie Tuner. The algorithm chooses items based on two objectives: 1) the items should be sufficiently different along the critique dimension, and 2) the items should be similar overall to the original item. We first define an objective measure of *critique distance*, the difference between items along the critique dimension. We then define a measure of the similarity between items. Finally, we present an algorithm that chooses items based on satisfying these two metrics simultaneously.

5.1 Critique distance

Users specify the direction of their critiques, but the system must determine how far to move in that direction. For example, if a user asks for a movie with less action than *Independence Day*, the system must decide whether to choose a movie like *Star Trek: Generations*, which still has a reasonable amount of action, or a movie like *Contact*, which has very little action.

[2] 0.325 is the Shannon entropy of the distribution $\{0.9, 0.1, 0.0\}$

To formalize these concepts, we introduce a metric called *critique distance* that measures the difference in tag relevance between two items with respect to a particular critique. For example, if a user applies the critique "less action" to *Independence Day*, then the critique distance to *Star Trek: Generations* is $\mathrm{rel}(\textit{Independence Day}, \textit{action}) - \mathrm{rel}(\textit{Star Trek: Generations}, \textit{action})$ = 0.97 − 0.46 = 0.51. Formally, if i_c is the critiqued item, i_r is the retrieved item, and (t, d) is the critique with tag $t \in T$ and direction $d \in \{-1, +1\}$, then

$$\text{critique-dist}(i_c, i_r, t, d) = \max(0, (\mathrm{rel}(i_r, t) - \mathrm{rel}(i_c, t)) \cdot d)$$

To determine the appropriate critique distance when choosing new items, we define a *critique satisfaction* metric that determines how strongly an item satisfies a critique based on critique distance. Below we define two alternative critique satisfaction metrics: $\mathrm{linear\text{-}sat}$ and $\mathrm{diminish\text{-}sat}$. In Section 5.3 we describe how we use these critique satisfaction metrics in conjunction with item similarity to sort critique results.

linear-sat: The *linear* critique satisfaction model, $\mathrm{linear\text{-}sat}$, assumes that critique satisfaction is proportional to critique distance. This model assumes that greater critique distance is always better, and that the rate of improvement stays constant as the critique distance increases. This model suggests that users want to move as far as possible along the critique dimension.

Formally, if i_c is the critiqued item, i_r is the retrieved item, and (t, d) is the critique, $t \in T, d \in \{-1, +1\}$, then

$$\text{linear-sat}(i_c, i_r, t, d) = \text{critique-dist}(i_c, i_r, t, d)$$

diminish-sat: The *diminishing returns* model, $\mathrm{diminish\text{-}sat}$, also assumes that greater critique distance is better, but that the rate of improvement decreases as the critique distance increases. This model suggests that users want a certain amount of change along the critique dimension, but that differences beyond that threshold have little value. Formally,

$$\text{diminish-sat}(i_c, i_r, t, d) = 1 - e^{-5 \cdot \text{critique-dist}(i_c, i_r, t, d)}$$

This formula is based on the negative exponential utility function. We chose the value of -5 based on qualitative analysis over a series of 30 test cases; this value tended to produce critique results that were noticeably different along the critique dimension, but not as different as those generated from the linear model.

5.2 Item similarity

When responding to a critique, the system should choose items that satisfy the critique, but are otherwise similar to the original item. One approach for measuring similarity between items is to use a domain-specific similarity metric; for example, in movie recommenders like MovieLens, a common similarity metric is the ratings correlations between movies. Alternatively, one could measure similarity of items based on the similarity of their tag genomes. We prefer the latter approach because it is domain-independent, and because the dimensions (i.e. tags) used to critique items are the same ones used to assess similarity. This means that items will tend to be similar along the dimensions visible to users.

We define the similarity between items i and j as the weighted cosine similarity of their tag genomes $\mathrm{rel}(i)$ and $\mathrm{rel}(j)$ (see Section 3.1). We used a weighted version of cosine similarity to account for the fact that some tags may be more important than others in determining similarity between items.

We denote the weighted cosine similarity between two vectors $\boldsymbol{x}$ and $\boldsymbol{y}$ based on weight vector $\boldsymbol{w}$ as

$$\text{cosine}(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{w}) = \frac{\sum_{k=1,\ldots,n} w_k \cdot x_k \cdot y_k}{\sqrt{(\sum_{k=1,\ldots,n} w_k \cdot x_k^2)} \cdot \sqrt{(\sum_{k=1,\ldots,n} w_k \cdot y_k^2)}}$$

We assign weights to tags based on two criteria: *tag popularity* and *tag specificity*. We assign more weight to popular tags because they reflect dimensions that more users care about. We define tag popularity of tag t as the number of users who have applied t, denoted as $\text{tag-pop}(t)$. We apply a log transform to make the distribution more normal.

We also assign higher weight to tags that are more specific, because specific tags can more uniquely identify similarities between items. For example, if two movies have the tag *dark comedy* in common, they are more likely to be similar than if they simply had the tag *comedy* in common. We measure tag specificity based on a modified version of *inverse document frequency*, a metric used in the tf-idf weighting scheme to assess term specificity [11]. In our case, we define $\text{doc-freq}(t)$ as the number of items where the relevance of t is greater than ½. We apply a log transform to the document frequency to bring it closer to a normal distribution.

Putting all of this together, we define the similarity between items i and j as

$$\mathrm{sim}(i, j) = \text{cosine}(\mathrm{rel}(i), \mathrm{rel}(j), \boldsymbol{w}),$$
$$\text{where } w_k = \frac{\log(\text{tag-pop}(t_k))}{\log(\text{doc-freq}(t_k))}$$

For all of the computations below, we normalize similarity values by subtracting the average similarity of all item pairs (0.61). Normalizing in this way yields similarity values with greater proportional variation, which helps balance the effects of similarity versus critique distance in the algorithm discussed in the next section.

5.3 Algorithm for responding to critiques

We now describe an algorithm that uses the above metrics to choose items in response to user critiques. Our general approach is to display a small set of highly relevant results, but let users explore a larger result set if they wish. The

interface design reflects this approach: critique results are displayed in a scrollable window sorted in descending order of goodness-of-fit to the critique, as shown in Figure 1.

The algorithm has two steps: a *filtering* step that establishes the basic requirements for an item to be included in the critique results, and a *sorting* step that orders the remaining items in descending order of goodness-of-fit to the critique.

Filtering. As discussed above, the system must choose items that are sufficiently different along the critique dimension, but are similar overall to the original item. Accordingly, the algorithm filters items based on both objectives. Filtering by similarity has the added benefit that it reduces the number of items to evaluate when responding to critiques of a particular item, enabling the data to be stored client-side.

- **Filtering based on critique distance.** Given a critiqued item i_c, tag t, direction $d \in \{-1, +1\}$, any result i_r must satisfy $\text{critique-dist}(i_c, i_r, t, d) > 0$.
- **Filtering based on overall similarity.** Given a critiqued item i_c, any result i_r must be among the k-nearest neighbors of i_c, based on the similarity metric defined in 5.2. We considered setting a minimum similarity value instead of using similarity rank, but found that the range of similarity scores varied between items. We chose a value of $k = 250$, because items outside that range tended to be considerably different from the critiqued item, and this value produced sufficiently long results lists to satisfy most users.

Sorting. The goal of the sorting step is to identify the items that most strongly satisfy the critique based on critique distance and are most similar to the original item. We sort results using a metric called *critique fit* that combines both objectives:

Given a critiqued item i_c, retrieved item i_r, tag t, direction $d \in \{-1, +1\}$,

$$\text{critique-fit}(i_c, i_r, t, d) = \text{critique-sat}(i_c, i_r, t, d) \cdot \text{sim}(i_c, i_r)$$

We implemented two versions of the sorting algorithm, one where critique-sat = linear-sat, and one where critique-sat = diminish-sat. The choice of function determines the tradeoff between critique distance and overall similarity. When critique-sat = linear-sat, the tradeoff between critique distance and overall similarity is the same at any critique distance. When critique-sat = diminish-sat, the tradeoff favors increased similarity over increased critique distance as critique distance increases. Table 2 shows sample results for both versions of the algorithm.

Compound and null critiques. The above definitions applies to unit critiques. For compound critiques, we simply take the product of the critique fit values for each of the individual critiques. When no critique has been applied, we order results based on similarity only.

Linear	Diminishing Returns
Aladdin (1992)	Shrek (2001)
Sword in the Stone (1963)	Shrek 2 (2004)
Toy Story (1995)	Toy Story 2 (1999)
Robin Hood (1973)	Aladdin (1992)
Looney, Looney, Looney Bugs Bunny Movie (1981)	Shrek the Halls (2007)

Table 2: Top-5 results for the critique "more classic than *Shrek the Third*" for both versions of the algorithm. The linear model favors more classic movies while the diminishing returns model favors similar movies.

6. DESIGN OF FIELD STUDY

We conducted a field study of Movie Tuner on the MovieLens website, in which we empirically evaluated Movie Tuner based on activity logs and survey data. The primary data source for our analyses comprised activity logs collected during a 7-week period that Movie Tuner was in place, running from July 14, 2010 through September 1, 2010, and the 7-week period just before the launch. These logs track all activity on Movie Tuner, including page views[3], critiques applied, and items selected.

We used a between-subjects design so that subjects could respond to survey questions based on their overall experience in a single experimental condition. Each user was assigned to one of four experimental groups based on two manipulated factors (2x2). One factor determined how Movie Tuner selected tags to display for an item: specifically, whether the algorithm favored *descriptive* tags (rel metric) or *discriminating* tags (critique-entropy metric), as described in Section 4.5. The other factor determined how Movie Tuner chose items in response to a critique: specifically, whether the algorithm used the linear (linear-sat) or the diminishing returns (diminish-sat) model of critique distance, as discussed in Section 5.1.

On 08/26/10, we invited 910 Movie Tuner users to an online survey, of whom 160 (18%) participated. We included users who had viewed Movie Tuner at least once and consented to participate in studies on MovieLens. In the survey, users responded to a series of statements, summarized in Table 3, using a 5-point Likert scale[4]. For each statement, we showed subjects a screenshot of the Movie Tuner interface for the movie *Pulp Fiction*, in order to help them recall their experience with Movie Tuner. We recognized that users may be influenced by the example shown to them when answering questions; therefore we displayed screenshots for each subject that matched how the interface would look given their experimental group. Additionally, we emphasized to subjects that they should respond based on *their* experience with Movie Tuner.

[3]We did have one logging problem during the time of the experiment. We did not lose any Movie Tuner data, but due to a data collection bug, movie detail page view data for pages that did not include Movie Tuner was lost between 07/22/10 and 7/29/10, and between 08/17/10 and 08/30/10. We believe this page view data would have been similar to the page view data that was correctly collected, so the lost data should not substantively affect the results.

[4]1 = *strongly disagree*, 2 = *disagree*, 3 = *neutral*, 4 = *agree*, 5 = *strongly agree*

Statement (abbreviated)	μ	**% agree**	**% disagree**
I would like the Movie Tuner feature to remain.	4.2	79	6
Movie Tuner is fun to use.	3.9	74	9
I like having the ability to specify critiques.	4.3	89	4
The tags shown helped me learn about the movie.	3.5	59	12
I liked seeing the tags.	3.9	72	6
The tags made sense to me.	4.0	81	8
The similar movies helped me discover movies I had not seen.	3.4	54	22
The similar movies helped me find movies I was interested in.	3.8	67	10
The similar movies were actually similar to the main movie.	3.6	60	7
Applying critiques helped me to discover movies I had not seen.	3.5	65	19
Applying critiques helped me find movies I was interested in.	3.7	71	11
Movies displayed in response to my critiques made sense.	3.8	68	8

Table 3: Survey questions and aggregated responses (5-point Likert scale). Percent (dis)agree equals the number of *(dis)agree* or *strongly (dis)agree* responses divided by total number of responses. For questions below the double line, we only included responses from users who actually applied critiques (71% of respondents.)

7. RESULTS

During the 7-week field trial, 2,531 users viewed the Movie Tuner interface a total of 49,099 times, and 1,037 users applied a total of 12,298 critiques. Overall feedback on MovieLens was positive; 89% of survey respondents liked being able to apply critiques, 74% found Movie Tuner fun to use, and 79% wanted Movie Tuner to remain available on MovieLens. Daily page views of the movie detail page increased by 52% ($p < 0.001$, t-test). One user commented, *"The best thing to come by in MovieLens (besides the product itself). Strongly recommended this to my friends and some picked MovieLens up just because of this addition. Love it!"*

In this section we empirically evaluate users' interactions with Movie Tuner, based on activity logs and user self-report. We first examine how users apply critiques in Movie Tuner, based on the types of tags they choose, how they choose critique direction, and whether they use compound or unit critiques. We then explore how users interact with items displayed in response to their critiques.

7.1 Applying critiques

Choosing tags. For 91% of critiques, users chose system-suggested tags rather than entering their own tags. This is consistent with interaction models suggesting people prefer recognition over recall [16]. Besides facilitating critique application, the system-selected tags provided other benefits: 72% of respondents like seeing the tags in Movie Tuner and 59% said the tags helped them to learn about the movie (compared to 12% who felt the tags did not help them learn).

Top 10 positive	**frac**	**Top 10 negative**	**frac**
nudity (full frontal -	0.19	coen brothers	0.08
nudity (full frontal)	0.15	religion	0.07
sexuality	0.11	holocaust	0.07
scary	0.09	world war ii	0.06
nudity (topless)	0.09	christmas	0.06
lesbian	0.08	western	0.06
black comedy	0.08	pixar	0.05
psychological	0.07	suicide	0.05
dark comedy	0.07	vampires	0.05
cyberpunk	0.07	police	0.04

Table 4: System-suggested tags most likely to be used in each critique direction, based on the fraction of times the tags were displayed that users chose them for critiques.

As discussed in Section 4.5, we implemented two algorithms for choosing tags, one that favored descriptive tags and one that favored discriminating tags. Survey results show that more subjects in the descriptive-tags group (87%) felt the system-suggested tags made sense to them compared to subjects in the discriminating-tags groups (74%). The differences are statistically significant both in percent agreement ($p < 0.05$, Z-test of proportions) and mean response ($p < 0.05$, t-test). We found no other statistically significant differences in survey responses between the two groups.

We compared critiques applied by users in each group, and we found that users in the discriminating-tags group chose a positive ("more") direction for 71% of their critiques, compared to 66% for users in the descriptive-tags group ($p < 0.01$, Z-test of proportions). This may be explained by the fact that the tags displayed by the descriptive-tags algorithm had a mean relevance of 0.81 to the movie displayed, while those displayed by the discriminating-tags algorithm had a mean relevance of only 0.48. As we will discuss later, users were more likely to apply critiques in a positive direction when tag relevance was low. However, we found no significant differences in the number of critiques applied or the proportion of users who applied critiques in the two groups.

With the exception of the differences described above, users in the two tag-selection groups exhibited similar behavior and expressed approximately the same level of satisfaction with Movie Tuner. This shows that Movie Tuner can support a range of tag-selection algorithms, and system designers may wish to explore algorithms that incorporate other objectives. For example, a system might choose a set of tags that capture a range of moods, or it might choose tags that steer users toward items that are otherwise hard to find.

Table 4 shows the system-suggested tags users were most likely to choose in each critique direction[5]. For positive critiques, many of the top-10 tags had sexual themes; for neg-

[5]Based on the number of times users applied the tag in that direction divided by the number of times the tag was displayed. We only included tags displayed at least 100 times

Top 10 positive	count	Top 10 negative	count
nudity	36	comedy	21
comedy	32	violence	9
mystery	30	violent	8
nudity (topless)	29	drugs	5
romance	29	horror	5
sex	28	sex	5
action	26	cheesy	4
surreal	21	dark	4
funny	18	nudity	4
erotic	16	predictable	4

Table 5: User-entered tags most frequently used in each critique direction.

ative critiques, many of the tags described sensitive topics such as *religion*, *holocaust*, or *suicide*.

Users entered their own tag rather than choose a system-selected tag for 9% of critiques. Table 5 shows the most popular user-entered tags for each critique direction. For positive critiques, the top-10 tags reflect similar themes to what we saw for the system-selected tags. For negative critiques, several tags mirror the criteria used to determine MPAA ratings (*violence*, *drugs*, *sex*, *nudity*), suggesting that some users are seeking to avoid movies with content they consider objectionable, perhaps because they wish to find movies appropriate for a younger audience. In both directions, the user-entered tags appear to be more general than the system-selected tags, suggesting that users may find it easier to recognize specific tags than to recall them.

Choosing direction. Users applied 68% of their critiques in the positive ("more") direction, compared with 32% in the negative ("less") direction ($p < 0.001$, Z-test of proportions). We expected that users would be more likely to select "more" for low-relevance tags compared to high-relevance tags, since there is greater distance to travel in the positive direction. To test this hypothesis, we divided critiques into three buckets based on the relevance of the critique tag t to the critiqued item i: *low relevance* ($\mathrm{rel}(i,t) < \frac{1}{3}$), *medium relevance* ($\frac{1}{3} \leq \mathrm{rel}(i,t) < \frac{2}{3}$), and *high relevance* ($\mathrm{rel}(i,t) \geq \frac{2}{3}$). The proportion of positive critiques in each bucket were 70.2%, 69.7%, and 66.1% respectively; the differences between the high relevance bucket and the other buckets were statistically significant ($p < 0.01$, Z-test of proportions), but the difference between the low and medium relevance buckets were not. These results show that lower tag relevance does correspond with a greater frequency of positive critiques, but that the effect is fairly weak. Among critiques with $\mathrm{rel}(i,t) > 0.95$, users still chose a positive direction 66% of the time. Future research should explore why users choose positive critiques most of the time: is it because tags tend to reflect attributes that people like, or because users find it more natural to navigate in a positive direction?

Unit versus compound critiques. Compound critiques were popular, comprising 24% of all critiques applied. 37% of users who applied a critique applied at least one compound critique. Several subjects who didn't realize compound critiquing was available asked for the feature in their comments. One subject wrote, *"I would like the Movie Tuner to permit adjusting two or more qualities at the same time. For example, if I am at the tuner for the movie 'The Girl Who Played with Fire', I would like to be able to search for movies that are both 'less violent and less sexually graphic'."*

7.2 Critique results

We also analyzed how users interacted with the results that Movie Tuner displayed in response to their critiques. On average, users clicked on 1.2 results for every movie they critiqued[6]. Survey results indicate that users were satisfied with the critique results: 68% of subjects who applied critiques thought the critique results made sense, 71% felt that applying critiques helped them find movies they were interested in, and 65% thought that applying critiques helped them find movies they had not seen. To make it easier to find movies they had not seen, several users asked for the option to exclude movies they had already rated.

We compared how users in the *linear* and *diminishing-returns* groups (see Section 5.1) responded to critique results. We found no statistically significant differences between the groups based on user self-report or observational data such as number of click-throughs. This suggests that Movie Tuner can support a range of approaches for responding to users' critiques. System designers may wish to incorporate other objectives when choosing items in response to critiques, such as predicted item rating or diversity of items.

Besides displaying movies in response to users' critiques, Movie Tuner also displays an initial list of "similar movies" when the user first visits a movie page. This feature proved popular: users clicked on the "similar movies" 12,626 times. Survey results indicate that users liked seeing the similar movies: 67% of subjects thought the similar movies helped them find movies they were interested in, and 54% thought it helped them to find movies they hadn't seen (versus 22% who did not think it helped find movies they hadn't seen). Further, 60% thought that the movies shown were actually similar to the main movie (versus 7% who did not), suggesting that the similarity metric worked properly.

8. CONCLUSION

In this paper we introduced Movie Tuner, a system for navigating an information space using natural language critiques based on community tags. In contrast to traditional tag search, Movie Tuner lets users formulate their preferences adaptively by critiquing particular examples. In contrast to traditional example-critiquing systems, Movie Tuner builds its knowledge base automatically by applying machine learning to user-contributed content, rather than by relying on paid experts.

We approached this problem from a design perspective, exploring two design dimensions. First, we examined how the system should suggest tags to users. We implemented two algorithms, one that favored *descriptive* tags and one that

[6]This only includes results clicked while a critique was in place.

favored *discriminating* tags. Survey participants felt that descriptive tags made more sense to them than discriminating tags. Users who saw descriptive tags tended to apply fewer positive critiques, most likely because descriptive tags represented attributes that were already fully present in the current item. Second, we explored how to choose items in response to users' critiques. We implemented *linear* and *diminishing returns* models of critique satisfaction based on critique distance. We found that users were equally satisfied and exhibited similar behavior with both approaches.

Initial tests suggest that Movie Tuner is an important and valuable tool. 89% of subjects liked being able to critique movies, and 79% wanted Movie Tuner to remain available on MovieLens. One user wrote, "*Movie Tuner instantly made MovieLens many times more valuable and useful for me! It generally works well and sometimes extremely well. Please keep it available!*" Over 1,000 users applied a total of 12,000 critiques, and views of movie detail pages on MovieLens increased by over 50%.

Since the results show that Movie Tuner may support a range of implementations, we encourage system designers to explore alternate designs. For example, some users suggested they would like more fine-grain control over their critiques. One user wrote, "*As opposed to a less/more function, the ability to slide the bar and set an amount would be welcomed.*" Alternatively, system designers may explore more organic implementations such as speech-based interfaces; for example, someone listening to a personalized radio station could simply say "less classical" or "more mellow" to select a song that better fits their mood.

One key advantage of the "Tuner" approach to building tag browsing applications is that the necessary information is contributed by users, and extracted by automated machine learning techniques. "Tuners" can be created for any information space with sufficient training data along with a community that is willing to share their views of the relationships between tags and items. Many applications are possible, including hunting for an apartment, reading the news, and even finding new friends. System designers will want to choose a set of training data suitable for their domain. For Movie Tuner, we found that text reviews provided the richest data for learning the tag genome, but tuners may also utilize other types of media. For example, a Tuner for music might extract features such as tempo, volume, and pitch from audio tracks to help learn the relevance of tags such as *relaxing*, *upbeat*, or *jarring*. Designers may experiment with algorithms tailored to the needs of each application.

9. ACKNOWLEDGMENTS

This paper is funded in part by National Science Foundation grants IIS 03-24851, IIS 05-34420, IIS 09-64695, and IIS 09-64697.

10. REFERENCES

1. R. D. Burke, K. J. Hammond, and B. C. Young. The FindMe approach to assisted browsing. *IEEE Expert*, 12:32–40, 1997.
2. L. Chen and P. Pu. Evaluating critiquing-based recommender agents. In *In Proc. AAAI 2006*, pages 157–162, 2006.
3. B. Faltings, P. Pu, M. Torrens, and P. Viappiani. Designing example-critiquing interaction. In *IUI '04*, pages 22–29, New York, NY, USA, 2004. ACM.
4. R. Fletcher. *Practical Methods of Optimization: Vol. 2: Constrained Optimization*. John Wiley and Sons, 1981.
5. S. J. Green, P. Lamere, J. Alexander, F. Maillet, S. Kirk, J. Holt, J. Bourque, and X.-W. Mak. Generating transparent, steerable recommendations from textual descriptions of items. In *RecSys '09*, pages 281–284, New York, NY, USA, 2009. ACM.
6. Y. Kammerer, R. Nairn, P. Pirolli, and H. Chi. Signpost from the masses: Learning effects in an exploratory social tag search browser. In *CHI '09*, pages 625–634.
7. G. Linden, S. Hanks, and N. Lesh. Interactive assessment of user preference models: The automated travel assistant. In *User Modeling '97*, pages 67–78. Springer, 1997.
8. K. McCarthy, J. Reilly, L. McGinty, and B. Smyth. Experiments in dynamic critiquing. In *IUI '05*, pages 175–182, New York, NY, USA, 2005. ACM.
9. S. Niwa, T. Doi, and S. Honiden. Web page recommender system based on folksonomy mining. In *ITNG '06*, pages 388–393, Washington, DC, USA, 2006. IEEE Computer Society.
10. J. W. Payne, J. Bettman, and E. J. Johnson. *The Adaptive Decision Maker*. Cambridge University Press, 1993.
11. G. Salton and M. McGill. *Introduction to Modern Information Retrival*. McGraw-Hill, 1983.
12. S. Sen, F. M. Harper, A. LaPitz, and J. Riedl. The quest for quality tags. In *GROUP '07*, pages 361–370, New York, NY, USA, 2007. ACM.
13. S. Sen, S. K. Lam, A. M. Rashid, D. Cosley, D. Frankowski, J. Osterhouse, F. M. Harper, and J. Riedl. tagging, communities, vocabulary, evolution. In *CSCW '06*, pages 181–190, New York, NY, USA, 2006. ACM.
14. S. Sen, J. Vig, and J. Riedl. Tagommenders: Connecting users to items through tags. In *WWW '09*, pages 671–680, New York, NY, USA, 2009. ACM.
15. C. Shirky. Ontology is overrated. http://www.shirky.com/writings/ontology_overrated.html, 2005. Retrieved on May 26, 2007.
16. D. C. Smith, C. Irby, R. Kimball, B. Berplank, and E. Harslem. Designing the Star user interface. *Human-computer interaction*, pages 237–259, 1990.
17. B. Smyth, L. McGinty, J. Reilly, and K. McCarthy. Compound critiques for conversational recommender systems. In *Web Intelligence '04*, pages 145–151, Washington, DC, USA, 2004. IEEE Computer Society.
18. J. Vig, S. Sen, and J. Riedl. Computing the tag genome. Technical report, University of Minnesota, 2010. http://www.grouplens.org/system/files/genome.pdf.
19. J. Zhang and P. Pu. A comparative study of compound critique generation in conversational recommender systems. In *Adaptive Hypermedia '06*, pages 234–243. Springer, 2006.