Generating Interesting Song-to-Song Segues With Dave

Giovanni Gabbolini
Insight Centre for Data Analytics
School of Computer Science & IT
University College Cork, Ireland
giovanni.gabbolini@insight-centre.org

Derek Bridge
Insight Centre for Data Analytics
School of Computer Science & IT
University College Cork, Ireland
derek.bridge@insight-centre.org

Abstract

We introduce a novel domain-independent algorithm for generating interesting item-to-item textual connections, or segues. Pivotal to our contribution is the introduction of a scoring function for segues, based on their 'interestingness'. We provide an implementation of our algorithm in the music domain. We refer to our implementation as Dave. Dave is able to generate 1553 different types of segues, that can be broadly categorized as either informative or funny. We evaluate Dave by comparing it against a curated source of song-to-song segues, called The Chain. In the case of informative segues, we find that Dave can produce segues of the same quality, if not better, than those to be found in The Chain. And, we report positive correlation between the values produced by our scoring function and human perceptions of segue quality. The results highlight the validity of our method, and open future directions in the application of segues to recommender systems research.

Keywords

segues, user studies, recommender systems, interestingness

1 Introduction

Often a recommender system makes use of relationships between items. Knowledge of these relationships and can be used in the recommendation task itself and can enable the explanation of the recommendations. However, there remains much room for research into the discovery of these relationships and how to measure their strength. In [3], Behrooz et al. propose the concept of "segues". Segues are short texts that explicitly connect one item to another. In [3], segues are used as a mean of enhancing the user experience with voice assistants in music streaming services. The authors provide a simple prototype which is able to generate song-to-song segues, and they evaluate the prototype qualitatively, with the goal only of exploring the potential of the idea.

In this paper we elaborate on the concept of segue, and we provide a domain-independent algorithm for generating interesting item-to-item textual connections. A distinguishing feature of our work is its ability to highlight interesting segues, according to a novel theory of interestingness.

Our algorithm assumes a knowledge graph as an abstract representation for items and information about items. In our abstraction, segues are paths from one item to another, and *interestingness* is a scoring function for paths. We 'get back' from the abstraction by mapping paths to texts.

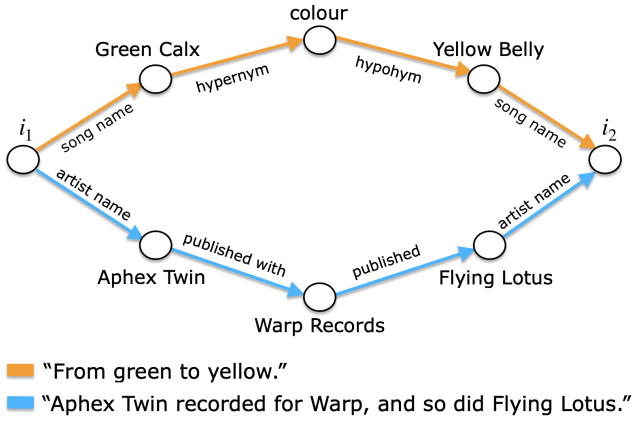

Figure 1: Examples of segues from the song "Green Clax" by "Aphex Twin" to the song "Yellow Belly" by "Flying Lotus", represented in the form of paths and texts.

We implement the algorithm in the music domain. We refer to our implementation as Dave. Dave can generate song-to-song segues of 1553 different types. Segue types range from factual to word-play. See Figure 1 for examples.

We evaluate Dave qualitatively by means of a user trial, where we compare Dave's segues against curated segues from a segment of the Radcliff & Maconie Show on BBC Radio 6 called The Chain. In the case of factual segues, we find that Dave can produce segues of the same quality, if not better, than those to be found in The Chain. The results highlight the validity of our method, and open future directions in the applications of segues to recommender systems research. We release the source code and the dataset, consisting of the answers gathered during the user trial, to facilitate future research on the subject.[1]

The remainder of this paper is organised as follows: Section 2 frames segues in the literature on recommender systems and user interfaces for music retrieval, and provides an overview of interestingness measures in data mining. Section 3 explains how Dave works. Section 4 details the experimental procedure and analyses the results. Section 5 discusses conclusions and future work.

UMAP '21, June 21–25, 2021, Utrecht, Netherlands

xxx-x-xxxx-xxxx-x/xx/xx…$xx.xx
https://doi.org/xx.xxxx/xxxxxxx.xxxxxxx

[1] https://github.com/GiovanniGabbolini/dave

2 Related Work

2.1 Explanations and recommender systems

Explanations of recommendations is an active research topic that has helped to increase the value of recommender systems, guided by goals that go beyond recommendation accuracy [36]. We refer the reader to the paper by Nunes and Jannach [27] for a comprehensive survey on the subject. Vig et al. [39] divide explanations into three categories: feature-based ("We have recommended this item because of your interest in this feature"), item-based ("We have recommended this item because you like this other item") and user-based ("People who liked this item also liked this other item"). Segues are strongly related to both item-based and feature-based explanations. In fact, segues make item-to-item connections explicit by leveraging item features. They are therefore related to the item- and feature-based hybrid explanations described in [30].

There is a growing body of research that uses knowledge graphs for computing recommendations [4, 7, 8, 22–24, 28] and some research that then uses the knowledge graph for explanations of those recommendations [5, 18, 25, 26, 31, 41]. For example, in [25, 26] the authors explain recommendations by jointly representing the following in a knowledge graph: items liked by a user; items recommended to the user; and features mined from DBPedia. The candidate explanations are paths from the user profile to the recommended items, which are ranked and translated into natural language. Bellini et al. [5] propose an autoencoder neural network for explainable recommendation. The hidden layer, and its connections to the input and output, are replaced by a knowledge graph. The graph is mined from DBPedia, and contains item features. In parallel with recommendations, the model also delivers explanations by leveraging the weights learned for the graph edges.

In some recommenders, the items to be recommended are inferred from paths, and the same paths are also used for explanations. Passant [31], for example, proposes a heuristic for scoring paths by relevance, and recommends items associated with the most relevant paths. In [18, 41], paths are scored through the use of deep neural networks that learn from past user interactions. We consider our work to be closely related to path-based recommenders: we introduce a scoring function for paths that can be used for recommendation and explanation purposes. Our scoring function is novel because it strives to maximize the quality of segues.

2.2 User interfaces for music retrieval

The study of intelligent user interfaces for music retrieval has been an active research topic for twenty years now [17]. The main goal of early interfaces was to facilitate the browsing of personal music collections, for example, by representing songs in 2D or 3D maps. Gulik and Vignoli [13] visualize artists as dots in a 2D map where proximity is determined by similarity across attributes such as genre, tempo and year; Tzanetakis et al. [38] visualize songs as dots in a 3D map, arranged by running PCA on song features, and where the genre of a song determines the colour of its dot. Another approach consists in representing collections linearly. Pohle et al. [32], for example, arrange songs circularly by solving a Travelling Salesman Problem, where the distances are determined by song-to-song similarity.

Recently, with the advent of streaming services, the quantity of music available to listeners shifted from relatively small personal collections to, virtually, all music content. Modern interfaces are, for the most part, not built to handle the visualization of these very large collections, but to automatically search for personalized sets of songs, and to show just these to the user [17]. In Pampalk et al. [29], for example, new artists are recommended based on a seed artist picked by the user. The recommendation algorithm is based on artist-to-artist similarity. The seed song is visualized as a sun, and the recommendations as sun rays, labelled by keywords.

The idea of segues is complementary to the work on user interfaces for music retrieval as proposed in the literature. These interfaces use similarity to facilitate browsing, search and recommendation [16]. Segues, on the other hand, make the connections explicit. Segues were first introduced in [3]. There, the authors developed a simple prototype able to generate segues for consecutive songs in playlists. They score individual segues simply with static segue preference weights but they also take into account, for example, the position of the segue in the playlist and its proximity to segues of the same type, in order to obtain a degree of segue diversity. Their experimental procedure consisted of unstructured interviews, aimed at exploring the potential of the idea. Our contribution fills a number of gaps in this previous work. We develop interestingness, a scoring mechanism for individual segues. And, we evaluate our system with a larger user trial, comparing it with a curated baseline.

2.3 Interestingness measures for data mining

Data mining is the discovery of patterns in large and complex datasets [14]. Measuring the interestingness of discovered patterns is an active and important field of study, surveyed by Geng and Hamilton [12]. Most research in this direction has focused on the interestingness of certain kinds of patterns, such as association rules, classification rules and summaries. Geng and Hamilton highlight that interestingness is usually defined as a combination of different components, such as: conciseness (rewarding patterns that contain a relatively small number of elements); peculiarity or infrequency (rewarding patterns if they are far away from other discovered patterns, according to some distance measure); surprisingness (rewarding patterns that contradict a person's existing knowledge or expectations); and actionability in some domain (rewarding patterns if they enable decision-making about future actions in this domain).

There is a smaller amount of research on the interestingness of paths in graphs. Lin and Chalupsky [19], for example, propose that the interestingness of a path of a given type is determined by the infrequency of that type of path in the graph. Ramakrishnan et al. [34] leverage three heuristics for interestingness. The first implements the idea that more specific nodes and edges convey more information than general ones, e.g. "singer" is more specific than "person". They compute specificity from hierarchies of node and edge types. Their second heuristic favours paths that cross different domains, e.g. from music to cinema. For this, a manual labelling, assigning domains to the nodes in the graph, is required. The third takes into account the infrequency of the path type. Aleman-Meza et al. [1] also introduce three heuristics for interestingness. The

first implements the idea of conciseness, and is computed from path length. The second is a heuristic for user-personalized paths. It leverages personalized weights assigned to regions of the graph. The third is concerned with trust in data sources. Again in this case, trust weights are manually assigned to regions of the graph.

3 Method

Our goal is to generate interesting item-to-item textual connections, or "segues". That is, we aim for interestingness, and we discourage trivial and boring segues. We are interested in generating factual connections, but also in amusing ones, based on simple word-play.

3.1 Algorithm

Our algorithm uses a knowledge graph G as an abstract representation for items and information about those items. In our abstraction, segues are paths from an item to another item, and interestingness is a scoring function from paths to numbers. We 'get back' from the abstraction through another function, which maps paths to texts. In particular, the algorithm for finding a segue from an item i_1 to another item i_2 works as follows: we find the paths in G that connect i_1 to i_2, we score them based on their interestingness, keeping only the best one, which is translated to text, and finally returned. This approach allows us to exploit heterogeneous data [11], and is domain-independent. The algorithm is summarized as Algorithm 1.

Algorithm 1: $find\_segue(G, i_1, i_2)$

input : knowledge graph G, items i_1 and i_2
output: interesting segue

```
paths = find_paths(G, i1, i2)
for path in paths do
    path.score = interestingness(path)
end
best_path = path with highest score
best_segue = path_to_text(path)
return best_segue
```

In the following, we will provide details for Algorithm 1. We start by presenting some preliminary concepts:

- A **knowledge graph** is a set of triples $G = \{(e, r, e') \mid e, e' \in E, r \in R\}$, where E and R denote, respectively, the sets of entities and relationships. A special subset of entities $I \subseteq E$, are the items (in our case, songs). Every entity has a *type* and a *value*. For example, an entity that represents a song has *type* equal to "song" and a *value* equal to the song URI. Every relationship has a *type*.
- A **path** p in G is an ordered list of entities and relationships in G, $p = [e_1, r_1, ..., r_{n-1}, e_n]$ where each triple in p must be in G. The *type* of p is the ordered concatenation of the entity and relationship types in p.

We turn to the components of Algorithm 1, namely *find_paths*, *path_to_text* and *interestingness*. *find_paths* is a path-finding procedure that returns all the simple paths, i.e. those without cycles, starting from the item i_1 and reaching the item i_2 in G . However, we allow for the possibility of constraining the paths that *find_paths* can find; see Section 3.2.2. *path_to_text* works by template filling, with canned templates indexed by the path type, and completed with information on the entities and relationships that constitute it. Our focus for this work was mainly on *interestingness* and so we devote the remainder of this section to explaining our definition.

Defining a scoring function for segues based on interestingness is not a trivial task. To be concrete, we refer to Figure 1: which one of the two segues is more interesting? There is no correct answer to this question, as interestingness can depend upon personal relatedness [35] and background knowledge [15]. In this context, we develop a simple theory of interestingness according to which a ranking can be determined. Our theory builds upon the concepts of infrequency and conciseness. We believe that infrequent segues are more interesting, as pointed out by Schank [35] and Kintsch [15] when discussing interestingess of general statements. We also believe that concise segues are more interesting, as supported by Geng et al. [12] in the context of interestingness in data mining. Our definition of interestingness applies to paths in knowledge graphs and consists of three heuristics. The heuristics rely only on statistical information and simple content descriptors, that can be defined independently of the domain, as they do not depend on the semantics of segues.

Rarity We define the rarity of a path p using the proportion of paths in G that have the same type as p. To formalize this, let T be the set of all path types in G; and let $f(t)$ be the number of paths in G that are of type t. Then,

$$rarity(p) = 1 - \frac{f(type(p))}{\max_{t \in T} f(t)}$$

Unpopularity We define the unpopularity of a path p using the notion of centrality of an entity e. An entity is central if it has a high number of incoming and outgoing edges, compared with other entities of the same type. A path that visits central entities is more popular than one that does not. To formalize this, let $edgeset(e)$ be the set of incoming and outgoing edges to and from an entity $e \in E$ in G. We define the centrality of an entity e as:

$$centrality(e) = \min\left(1, \frac{|edgeset(e)|}{\underset{e' \in E}{median}|edgeset(e')|}\right),\ type(e') = type(e)$$

Then, we define the *unpopularity* of a path p as:

$$unpopularity(p) = 1 - \min_{e \in p \cap E}(centrality(e))$$

Shortness Let the *length* of a path p in G be:

$$length(p) = |p \cap R|$$

We define the *shortness* of a path p in G as done in [2]:

$$shortness(p) = \frac{1}{length(p)}$$

The heuristics *rarity* and *unpopularity* both implement the idea of favouring infrequent segues, but in a different fashion: *rarity* has high values for infrequent path types, while *unpopularity* has high values for paths that include infrequent entities. For example, a path that connects people who share a birthday is likely to have a fairly high value for *rarity*, but not necessarily for *unpopularity*,

e.g. if both people are very famous. Or, paths that connect people who have the same hair colour will have a low value for *rarity*, but can have a high value for *unpopularity*, e.g. if the shared hair colour is something unusual such as "green". The *shortness* heuristic implements the concept of conciseness.

We define the *interestingness* of a path p in G as the convex combination of the three heuristics:

$$interestingness(p) = w_1 rarity(p) + w_2 unpopularity(p) + w_3 shortness(p)$$

It ranges from zero to one. w_1, w_2, w_3 are parameters to be tuned, subject to $w_1 + w_2 + w_3 = 1$.

3.2 Implementation

We implement Algorithm 1 as described in Section 3.1 for the music domain. We consider items to be songs, and obtain song-to-song segues as a result. We will refer to our implementation from now on as Dave.

3.2.1 Knowledge graph We represent a song with three fields: *song name*, *artist name* and *album name*. The representation allows missing values, e.g. a song that is not part of an album, and can be easily changed with only minor modifications to the rest of our implementation, e.g. if songs were instead represented by their Spotify URIs.

Our implementation uses a knowledge graph with 40 distinct node types and 230 distinct edge types, and can provide segues of 1153 different path types. We build the knowledge graph with data that we harvest from three main resources:

MusicBrainz We use MusicBrainz[2] as the main source of factual data. We exploit the MusicBrainz APIs.[3] They allow us to navigate the MusicBrainz database, and offer entity-linking functionalities. In a first step, we link the actual song, its album and its artist to their respective MusicBrainz URIs. Then, we mine different sorts of factual data, ranging from the genres of the song to the birth places of the artists.

Wikidata We use Wikidata[4] as an additional source of factual data. There exists a mapping from MusicBrainz URIs to Wikidata URIs, making it easy to use both resources. From Wikidata, we mine biographical data about artists that is not available in MusicBrainz, e.g. the awards that an artist has won.

WordNet We use WordNet [10] to mine lexical data about the words in song, artist and album names, e.g. hypernyms and hyponyms.

In addition, we gather some further lexical data through a number of different resources. For example, we link words in song, artist and album names to their stems using the Porter stemmer [33], and to their phonetics with the NRL algorithm [9]. We provide a complete description of entities and relationships that build up the knowledge graph in the additional materials.[5]

The factual data that we obtain from MusicBrainz and Wikidata allows for conventional, informative segues. On the other hand, the lexical data from WordNet and other resources yields less conventional and perhaps amusing connections based on word-play. We show two examples in Figure 1.

3.2.2 Algorithm As mentioned in Section 3.1, we constrain the *find_paths* component of Algorithm 1; specifically, we constrain it to find only paths that go from a start song to another song without visiting another song. This constraint limits the number of paths to be scored, at the price of losing some indirect paths. We believe that some of these indirect paths would be filtered out by *interestingness* anyway, since they would have low values for *shortness*. We set the weights to be used in the *interestingness* score to $w_1 = 0.4, w_2 = 0.2, w_3 = 0.4$. The weights were set after empirical experimentation, using songs that were not used in the user trial. We discuss other ways to set the weights in Section 5.

4 Experiments

We evaluate Dave by means of a user trial, where we compare Dave's segues against a curated source of segues called The Chain. The Chain is a segment of the Radcliffe & Maconie Show on BBC Radio 6. In this segment of the show, listeners call in and propose the next song, always making sure that there is a connection (sometimes informative, sometimes funny) between the previous song and the next one. The Chain is made for entertainment and therefore offers very interesting segues, with strong creative traits. A database with all the segues that have appeared in The Chain is available on the internet.[6] At the time of writing, it comprises more than 8000 segues. A direct comparison with curated segues is our means of assessing the quality of segues from Dave. We do not include any other algorithmic baseline, as we are not aware of any other similar systems in the literature. The only work that deals with segues is [3]. We cannot compare with their work their scoring mechanism is presented in insufficient detail to be reproducible.

Our user trial is a within-subject trial with two treatments: Dave and The Chain. We generate segues to show in each treatment using the following procedure. One segue is sampled at random from a fixed random sub-sample of The Chain, of cardinality equal to 200 segues. The corresponding segue is generated by Dave by picking the segue that maximizes the *interestingness*, when starting from the same song as The Chain, and going towards a song from a fixed sample of songs, of cardinality equal to 496 songs. These 496 songs are a random sub-sample of 20000 songs from the RecSys Challenge 2018 Dataset [6]. We use MusicBrainz as a source of artist data to filter out artists based on the country in which they were born or in which they are based, and on the musical genres with which they are associated. Specifically, we keep only artists born or based in the UK, and who are associated with at least one genre from the following: blues, blues-rock, pop, pop-rock, rock, soft-rock, funk, jazz, r&b and soul. This country and these genres are ones that predominate in The Chain. This filtering ensure that the songs that Dave can choose from match the style of songs found in The Chain, thus mitigating one confounder from the user trial.

Each participant is asked to evaluate six segues: she undergoes both treatments three times. The order of treatments in each pair

[2] https://musicbrainz.org/
[3] https://python-musicbrainzngs.readthedocs.io/en/v0.7.1/
[4] https://wikidata.org/
[5] https://doi.org/10.5281/zenodo.4619395
[6] https://www.thechain.uk

is randomized. We make sure that the same segues are not shown multiple times to a participant.

In the following, we provide details on the user trial design, we present statistics on the answers, and, finally, we analyze the results in depth.

4.1 User trial design

The user trial begins with an instructions page. It continues with a three-part survey, whose parts we will refer to as intro, segue evaluation and outro. The trial concludes with a final page that offers an optional comments box. In the following, we describe the three parts of the survey. We provide screenshots of the user trial text and its workflow in the additional materials.[7]

4.1.1 Intro In the intro, we ask each participant some questions to identify how much she engages with music. A previous study has highlighted that segues might be especially suited for music 'nerds' [3]. By asking these questions in our trial, we can see whether music engagement correlates with segue appreciation, for example. The source of the questions that we ask is the Goldsmiths Musical Sophistication Index (Gold-MSI) [21]. The index comprises five aspects. Four of the aspects are concerned with music skills, such as musical training and singing abilities. We restrict ourselves to the remaining aspect, the one called Active Engagement (AE). AE covers "a range of active musical engagement behaviours (e.g. I often read or search the internet for things related to music) as well as the deliberate allocation of time and money on musical activities (e.g. I listen attentively to music for *n* hours per day)" [21]. We follow [21], and we measure AE by asking ten questions, with answers on a seven point scale.

We are also interested in English proficiency, as we want to make sure that participants can properly understand the segues. So in the intro we also ask the participant to identify her level of proficiency from among four options: "Low", "Mid", "High" and "Mother Tongue". We do not collect demographic information, as we considered it not essential for the scope of the study.

4.1.2 Segue evaluation In the segue evaluation phase, each participant is asked to evaluate six segues, as described in Section 4. We show the segues, as well as the titles and artists of the songs that they connect. We do not provide any means for the user to listen to the songs. For every segue, we ask questions to measure a variety of quality metrics. First, we ask whether the segue is *likeable*, of *high-quality*, and whether it *sparked-interest* in the next song. Second, we asked how the participant perceived its content, on three dimensions: *informative*, *funny* and *creative*. Lastly, we wanted to make sure that the connection between the two songs is expressed in an *understandable* way by the segue, and that the segue is *well-written*. We call these three groups of dependent variables respectively: *valence*, *content* and *text* quality metrics. The dependent variables introduced in this part of the survey are summarized in Table 1. We measure all of them using five point Likert scales.

4.1.3 Outro In the outro part of the trial, we measure the familiarity of the participant with the songs and artists involved in the segues that she has evaluated. Familiarity with songs and artists might be a confounder for the quality metrics, and we want to address these effects, if any. We consider familiarity because it has been shown to be an accurate predictor of musical choice, at least as good as liking [40], and we assume there may be an extension of the results in [40] to segues. We measure familiarity with the songs and artists of both the first song and second song in each segue. Familiarity is measured with a simple two point scale: "Familiar" and "Not familiar". We decided to ask about familiarity in the outro, after the segue evaluation, so that we avoid accentuating any confounding effects.

[7] https://doi.org/10.5281/zenodo.4619395

Table 1: Dependent variables measured during the segue evaluation part of the user trial.

name	dependent variables
valence quality metrics	*likeable*, *high-quality*, *sparked-interest*
content quality metrics	*funny*, *informative*, *creative*
text quality metrics	*understandable*, *well-written*

4.2 Answer statistics

In total, 158 people completed the trial. They are undergraduate Computer Science students recruited in a university in Ireland. The median completion time for the survey is 7 minutes, with a maximum of 68 minutes and minimum of 2 minutes. We discard answers from people who took less than 3 minutes, as we are worried about their reliability. No participant declared their English proficiency to be "low" but we further filter out participants with "mid" English proficiency, since they also might not be able to properly evaluate segues. We are left with people with "high" proficiency and those for whom English is their mother tongue (90% of the total). After the filtering, we end up with 151 people. We convert Active Engagement answers to numbers from one to seven, and segue evaluation Likert scale answers to numbers from one to five. We also analyze the comments left by the participants, 35 comments in total, but we do not discuss them in this paper for lack of space and because they do not add much to our analysis.

User categories In some of our analysis, we partition participants based on their level of Active Engagement with music (AE). We summed the answers to the AE questions given by each participant, obtaining a distribution of total AE scores. We divided participants according to the quartiles of this distribution, giving four categories: "low-AE", "mid-low-AE", "mid-high-AE" and "high-AE". We validated this partition by considering confidence intervals of segue evaluation answers: the quartiles show good internal cohesion.

Segue categories We believe that segues can be divided into two categories: those that are intended to amuse ("funny") and those that are intended to impart information ("informative"). We decided to create a ground truth that assigns each segue to one of the two categories. We, the two authors of this paper, separately labeled every segue manually, guided by the following criterion: a segue is funny if it is written with the goal of making the listener smile, and a segue is informative if it is written with the goal of giving information to the listener. A segue can have both goals, e.g. if it presents information in a funny way. In such borderline cases, we

assigned the goal that seemed dominant. We disagreed upon the labelling in 13 out of 400 cases (Cohen's $k = 0.92$). We solved divergences as follows: given a segue s_1 where there was disagreement, we found a second segue s_2 whose category was not in dispute, and that both of us considered to be similar to s_1. We then assigned to s_1 the category of s_2. We validate the ground truth by double-checking it against the answers to the survey. In particular, participants were asked to express how much segues were *informative* and *funny*. We computed mean values of their answers, considering separately segues labelled as informative and funny in the ground truth. We carry out a t-test for assessing the significance of differences in the mean values. We found that the mean for *informative* is statistically significantly higher than the mean for *funny* for segues labelled as informative (3.64 vs 2.81, $p<0.001$), and the opposite for segues labelled as funny (2.47 vs 2.82, $p<0.001$). We conclude that participants in the survey agree with our manual labeling, thus providing some support for the reliability of the ground truth.

The ground truth reveals that The Chain is biased towards funny segues: roughly three out of every four of its segues are funny. Dave is approximately balanced. We show some examples of informative and funny segues in Tables 2 and 3.

4.3 Results

In this section, we analyze the answers to the segue evaluation part of the survey, dividing by treatments, segue category and user category. We also investigate the effect of familiarity. And, finally, we evaluate the effectiveness of *interestingness* and the correlation between quality metrics.

4.3.1 Performance in the quality metrics We compute the average for each quality metric given in Table 1 within treatment (Dave and The Chain), presenting separately the results for informative segues (Table 4) and funny segues (Table 5). We conduct a t-test for assessing the significance of differences between the two treatments. We discuss these results below.

Informative segues For informative segues (Table 4), Dave outperforms the human-curated segues of The Chain for two of the three *valence* quality metrics but the differences are not statistically significant. There are statistically significant differences on the *content* quality metrics: The Chain is perceived as more *funny* and *creative* ($p<0.05$), while Dave is more *informative* ($p<0.01$). Finally, turning to the *text* quality metrics, Dave's segues turn out to be better written and more *understandable* than The Chain's but again without statistical significance.

Funny segues When it come to funny segues (Table 5), human-curated segues from The Chain outperform Dave's segues across all the quality metrics, with statistically significant differences. We notice low values for *funny* in both treatments. We might expect it to be higher in the category of segues we are considering. This may be due to the medium of presentation of the segues, i.e. read on a screen. We might expect better results if, for example, segues were spoken. It may also just be that the word-play humour of these segues does not appeal to the sense-of-humour of the trial participants.

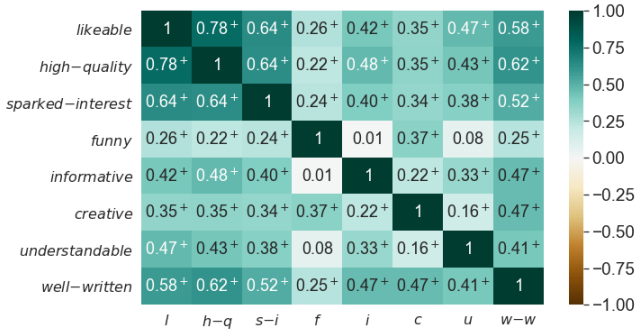

Figure 2: All segues. Quality metrics correlation matrix. $^+$: $p<0.001$, otherwise $p>0.05$

4.3.2 Correlation of quality metrics We compute pairwise correlations of the eight quality metrics and this is shown for all segues in Figure 2. There is high correlation (0.64, $p<0.001$) between both *high-quality* and *likeable* with *sparked-interest*: good segues can spark interest in the next song. We notice that *informative* has higher correlation than *funny* with all the *valence* quality metrics: a segue perceived as very informative is likely to be also perceived as very likeable, high-quality, and is more likely to spark interest in the next song. This happens to a lesser extent for segues perceived as very funny. Therefore, from a recommender systems point-of-view, where sparking interest is important, it might be more fruitful to address efforts into generating informative segues, as opposed to funny segues. However, this observation might be just due to the medium of presentation of the segues i.e. read. We do not know whether the result would generalize to other mediums, e.g. spoken.

The same happens with *understandable*, which is statistically significantly correlated with *informative* but not with *funny*. Further, *creative* has good correlation with all the *valence* quality metrics: creativity is a good asset for segues. Finally we report high correlation of *well-written* with all *valence* quality metrics (ranging from 0.52 to 0.62, $p<0.001$). This is expected, since segues are consumed in textual form. How they are written is very important, as important as the content itself.

We have also looked at these correlations in various subdivisions of the data: within treatment (Dave or The Chain), within segue category (funny or informative) and within user category (low-AE, etc.). The narrative that would accompany each of these correlation matrices is not appreciably different from the one we have given above. Hence, to save space, we do not show these more specific correlation matrices in this paper.

4.3.3 Performance in quality metrics and user category We divide users into four groups, as explained in Section 4.2, and we compute the average of the answers to the quality metric questions within each user category. We conduct a statistical test for assessing the significance of differences in performance, comparing the lowest level of AE with the other three. We show the results in Table 6. Only one metric changes statistically significantly: *sparked-interest*. This is reasonable: higher active engagement with music is somehow correlated with a propensity for music discovery. The other main quality metrics slightly increase with AE, but the differences are

Table 2: Examples of informative segues. Not only were these labeled informative in the ground truth but also at least two user trial participants gave them a maximum rating on the *informative* quality metric.

treatment	first song	segue	second song
Dave	*Weather To Fly* by *Elbow*	And now Guy Garvey, who was a member of Elbow...	*Belly Of The Whale* by *Guy Garvey*
The Chain	*One for the Road* by *Arctic Monkeys*	On Arctic Monkeys' last tour, Bill Ryder-Jones of The Coral joined them...	*Pass It On* by *The Coral*

Table 3: Examples of funny segues. Not only were these labeled funny in the ground truth but also at least two user trial participants gave them a maximum rating on the *funny* quality metric.

treatment	first song	segue	second song
Dave	*Fleety Foot* by *Black Uhuru*	From foot to faces…	*Faces* by *Ed Sheeran*
The Chain	*Tumbling Dice* by *The Rolling Stones*	You need a dice to play snakes and ladders...	*Rattlesnakes* by *Lloyd Cole and the Commotions*

Table 4: Informative segues. Average value of quality metrics achieved in the two treatments, on a scale from one to five. *: $p<0.05$; **: $p<0.01$.

	likeable	*high-quality*	*sparked-interest*	*funny*	*informative*	*creative*	*understandable*	*well-written*
Dave	3.26	3.22	3.06	2.39	3.73^{**}	3.33	3.84	3.46
The Chain	3.20	3.20	3.13	2.64^{*}	3.43	3.59^{*}	3.69	3.33

Table 5: Funny segues. Average value of quality metrics achieved in the two treatments, on a scale from one to five. *: $p<0.05$; **: $p<0.01$; ***: $p<0.001$.

	likeable	*high-quality*	*sparked-interest*	*funny*	*informative*	*creative*	*understandable*	*well-written*
Dave	2.94	2.77	2.76	2.71	2.66	3.22	3.58	3.13
The Chain	3.28^{***}	3.14^{***}	2.99^{*}	2.89^{*}	2.90^{**}	3.57^{***}	3.78^{*}	3.35^{*}

not statistically significant. The results we obtain if we further divide, e.g. by treatment or by segue category, confirm those that we have presented in Table 6. The results contradict the intuition that segues are especially suited to "nerds" [3]. The result might change if segues were to include more musicological detail, for example, the synthesizer brand used by two artists during a recording. At present, Dave's knowledge graph does not contains these kinds of details and so does not allow Dave to produce segues such as these.

4.3.4 Performance in quality metrics and familiarity We consider whether quality metrics are related or not to familiarity with the artists and songs involved in the segues. For lack of space, we focus our attention on familiarity with songs, stating only briefly the results we have for artists. We divide answers into four groups, based on whether participants are familiar or not with each of the two songs connected by the segue, and we compute the averages of each group. We conduct a statistical test for assessing the significance of differences in performance, comparing the first group (familiar with neither song) with the other three. We do not further partition by treatment, segue category or user category, as the cardinality of some of the groups is already small. We show the results in Table 7.

We observe that familiarity with songs, in general, leads to higher appreciation of segues. Segues are more *likeable* when connecting two familiar songs than when connecting two unfamiliar songs ($p<0.01$). And, segues are able to spark interest more when the songs are already familiar, with respect to when they are not ($p<0.001$). Moreover, they are perceived as better written ($p<0.01$), and more *understandable* ($p<0.05$).

When repeating the analysis but considering familiarity with the artists, we observe the same phenomena, but the increases in the metrics have lower magnitudes. We conclude that familiarity with artists is a weaker confounder than familiarity with songs.

4.3.5 Interestingness and quality metrics The *interestingness* function is our computational means for assessing whether a segue foud by Dave is good or not. We would like to verify whether it agrees with the human perception of quality or not. To this end, we compute the correlation of the *valence* quality metrics and *interestingness* for all of Dave's segues that were used in the user trial.

Table 6: All segues. Average value of quality metrics, divided by level of Active Engagement (AE) with music. Values range from one to five. We conduct a significance test, comparing the lowest value of AE with the other three. ***:** $p<0.05$; ****:** $p<0.01$; *****:** $p<0.001$.

	likeable	*high-quality*	*sparked-interest*	*funny*	*informative*	*creative*	*understandable*	*well-written*
low-AE	3.08	3.00	2.64	2.56	3.07	3.28	3.63	3.18
mid-low-AE	3.20	3.08	2.97^{**}	2.72	3.01	3.34	3.71	3.23
mid-high-AE	3.19	3.09	3.16^{***}	2.80^{*}	3.26	3.54^{**}	3.72	3.46^{**}
high-AE	3.25	3.16	3.09^{***}	2.66	3.19	3.53^{**}	3.86^{*}	3.42^{*}

Table 7: All segues. Average value of quality metrics, dividing answers based on the familiarity with the two songs connected by the segue. Values range from one to five. We conduct a significance test, comparing the first group (familiar with neither song) with the other three. ***:** $p<0.05$; ****:** $p<0.01$; *****:** $p<0.001$.

familiar with	*likeable*	*high-quality*	*sparked-interest*	*funny*	*informative*	*creative*	*understandable*	*well-written*
neither song	3.11	3.03	2.81	2.65	3.08	3.36	3.67	3.25
just 1^{st} song	3.22	3.08	3.08^{*}	2.64	3.30^{*}	3.45	3.87	3.40
just 2^{nd} song	3.29	3.18	3.27^{***}	2.79	3.16	3.57^{*}	3.86	3.41
both songs	3.49^{**}	3.29^{*}	3.47^{***}	2.78	3.28	3.61^{*}	3.93^{*}	3.61^{**}

We find that there is a statistically significant correlation between the *valence* quality metrics and *interestingness* for informative segues. This indicates that the *interestingness* function, without being aware of semantics, relying only on statistical information and simple content descriptors, can successfully rate the quality of informative segues: on average, segues rated low by the trial participants are low also in *interestingness*, and vice versa. We believe that this is a very good result, given the complexity of the task. We observe worse results with funny segues, where we do not find any statistically significant correlations: deeper considerations might be needed, e.g. the role of semantics. We also compute the correlation with *content* quality metrics, but we do not find any strong statistically significant correlations. This is expected, since *interestingness* is independent from the semantics of the segues.

Finally, we turn to the correlation with *text* quality metrics. We do not find any correlation between *interestingness* and *well-written*. This is as expected, since *well-written* does not depend directly on *interestingness*, but on *path_to_text*. But, we do find statistically significant correlation in informative segues for *understandable* (0.22, $p<0.001$). This is an indication that *interestingness*, even though it is built around the concept of infrequency, does not favour obscure segues.

We report all the results in Table 8.

Table 8: Dave's segues. Correlation of quality metrics and *interestingness* score. ***:** $p < 0.05$; ****:** $p < 0.01$; *****:** $p < 0.001$.

	interestingness	
	informative segues	funny segues
likeable	0.25^{***}	0.13
high-quality	0.23^{***}	0.11
sparked-interest	0.17^{**}	0.00
funny	-0.06	-0.17^{*}
informative	0.14^{*}	0.04
creative	0.15^{*}	-0.09
understandable	0.22^{***}	0.10
well-written	0.11	0.05

5 Conclusions and Future Work

In this paper we introduced a novel method for generating item-to-item segues and implemented it in the case of song-to-song segues in a system called Dave. Dave can provide a wide variety of segues, that can be categorized as either funny or informative. The core of our method is *interestingness*, a domain-independent function for scoring the interestingness of paths in knowledge graphs.

We evaluate Dave by means of a user trial, where we compare it against curated segues from a segment of the Radcliffe & Maconie Show on BBC Radio 6 program, called The Chain. The use of The Chain may be a limitation of this work. Segues from The Chain have a peculiar style that fits the radio program, and are tailored to a particular kind of audience. They tend to be amusing and very creative. Dave, on the other hand, tends to be factual, and can only deliver funny segues made of word-play. In order to alleviate such problems, we compared funny segues and informative segues from the two methods separately. Notice too that, even though The Chain has a high percentage of funny segues, this does not mean that its informative segues are weak: The Chain is curated by experts and draw on the considerable knowledge of thousands of BBC listeners. In any case, the user trial was intended as a method to assess how Dave's segues compare with curated segues from a trustworthy source — segues that we can assume to be "really good". This gives a way of finding how far Dave is from being "really good". Our goal is not to demonstrate that our algorithm can be substituted for the listeners to the show, instead, we aim to

provide an evaluation in a scenario where no algorithmic baseline is available.

We find that Dave can produce informative segues of the same quality, if not better, than The Chain. We believe that this is an astonishing result, that gives an idea of the quality of our method. But, when turning to funny segues, the results are not as good. We believe that this is partially due to the *interestingness* function, as we find evidence that it is much better suited to rate the quality of informative segues. Another reason might be that funny segues from Dave are limited to word-play, and this kind of humour does not appeal to everyone and may, in particular, not appeal to the participants in our user trial. In fact, even curated funny segues from The Chain are not perceived as funny by participants in our trial. This may be a mismatch in sense of humour between listeners to the show and participants in the trial. It may also be due, in part, to the fact that segues are being read rather than being spoken. It is fair to say that, overall, the task of tackling the funny segues is only partly solved by the proposed model.

We find that good segues can spark interest in songs, and that this effect increases the more the user has high active engagement with music. Future directions might include the construction of a recommender system that incorporates segues to guide the discovery of new music, especially helping to solve the open problem of acceptance of recommendations for songs that deviate from user expectations (so-called divergent song recommendations [20]). We also notice that segues are particularly appreciated when they deal with music the user is already familiar with. Future work might therefore also concern the application of segues to enhance recommendations for repeated consumption, a popular topic in the music domain [37].

We are also interested in exploring the idea of learning the weights used by the *interestingness* scoring function, and in particular learning personalized weights that would adjust to fit the tastes of the user.

Other future directions involve the use of multiple segues to provide a narrative to accompany a playlist.

Lastly, we are interested in exploiting the domain-independent nature of our contribution, and to work with other kinds of items. For example, segues might be used by point-of-interest recommenders in the tourism domain to build a story around places to visit, going towards the idea of a virtual and intelligent tour guide.

Acknowledgments

We thank Peter Knees of TU Wien for useful discussions about the design of the user trial.

This publication has emanated from research conducted with the financial support of Science Foundation Ireland under Grant number 12/RC/2289-P2 which is co-funded under the European Regional Development Fund. For the purpose of Open Access, the authors have applied a CC BY public copyright licence to any Author Accepted Manuscript version arising from this submission.

References

[1] Boanerges Aleman-Meza, Chris Halaschek, I. Budak Arpinar, and Amit Sheth. 2003. Context-Aware semantic association ranking. *Proceedings of the 1st International Conference on Semantic Web and Databases, SWDB 2003* (2003), 24–41.

[2] Boanerges Aleman-Meza, Christian Halaschek-Wiener, I. Budak Arpinar, Cartic Ramakrishnan, and Amit P. Sheth. 2005. Ranking complex relationships on the semantic web. *IEEE Internet Computing* 9, 3 (2005), 37–44. https://doi.org/10.1109/MIC.2005.63

[3] Morteza Behrooz, Sarah Mennicken, Jennifer Thom, Rohit Kumar, and Henriette Cramer. 2019. Augmenting Music Listening Experiences on Voice Assistants. In *Proceedings of the 20th International Society for Music Information Retrieval Conference, ISMIR 2019, Delft, The Netherlands, November 4-8, 2019*, Arthur Flexer, Geoffroy Peeters, Julián Urbano, and Anja Volk (Eds.). 303–310. http://archives.ismir.net/ismir2019/paper/000035.pdf

[4] Vito Bellini, Vito Walter Anelli, Tommaso Di Noia, and Eugenio Di Sciascio. 2017. Auto-Encoding User Ratings via Knowledge Graphs in Recommendation Scenarios. In *Proceedings of the 2nd Workshop on Deep Learning for Recommender Systems* (Como, Italy) *(DLRS 2017)*. Association for Computing Machinery, New York, NY, USA, 60–66. https://doi.org/10.1145/3125486.3125496

[5] Vito Bellini, Angelo Schiavone, Tommaso Di Noia, Azzurra Ragone, and Eugenio Di Sciascio. 2018. Knowledge-aware Autoencoders for Explainable Recommender Systems. *ACM International Conference Proceeding Series* (2018), 24–31. https://doi.org/10.1145/3270323.3270327 arXiv:1807.06300

[6] Ching-Wei Chen, Paul Lamere, Markus Schedl, and Hamed Zamani. 2018. Recsys Challenge 2018: Automatic Music Playlist Continuation. In *Proceedings of the 12th ACM Conference on Recommender Systems* (Vancouver, British Columbia, Canada) *(RecSys '18)*. Association for Computing Machinery, New York, NY, USA, 527–528. https://doi.org/10.1145/3240323.3240342

[7] Tommaso Di Noia, Roberto Mirizzi, Vito Claudio Ostuni, Davide Romito, and Markus Zanker. 2012. Linked Open Data to Support Content-Based Recommender Systems. In *Proceedings of the 8th International Conference on Semantic Systems* (Graz, Austria) *(I-SEMANTICS '12)*. Association for Computing Machinery, New York, NY, USA, 1–8. https://doi.org/10.1145/2362499.2362501

[8] Tommaso Di Noia, Vito Claudio Ostuni, Paolo Tomeo, and Eugenio Di Sciascio. 2016. SPrank: Semantic Path-Based Ranking for Top-N Recommendations Using Linked Open Data. *ACM Trans. Intell. Syst. Technol.* 8, 1, Article 9 (Sept. 2016), 34 pages. https://doi.org/10.1145/2899005

[9] H. S. Elovitz, R. W. Johnson, A. McHugh, and J. E. Shore. 1976. Automatic translation of English text to phonetics by means of letter-to-sound rules. Naval Research Lab. Report.

[10] C. Fellbaum. 2000. WordNet : an electronic lexical database. *Language* 76 (2000), 706.

[11] Dieter Fensel, Umutcan Şimşek, Kevin Angele, Elwin Huaman, Elias Kärle, Oleksandra Panasiuk, Ioan Toma, Jürgen Umbrich, and Alexander Wahler. 2020. *Introduction: What Is a Knowledge Graph?* Springer International Publishing, Cham, 1–10. https://doi.org/10.1007/978-3-030-37439-6_1

[12] Liqiang Geng and Howard J. Hamilton. 2006. Interestingness measures for data mining: A survey. *Comput. Surveys* 38, 3 (2006), 3. https://doi.org/10.1145/1132960.1132963

[13] Rob Gulik and Fabio Vignoli. 2005. Visual Playlist Generation on the Artist Map. In *in Proc. of the ISMIR Intl. Conf. on Music Information Retrieval*. 520–523.

[14] David J. Hand and Niall M. Adams. 2015. *Data Mining*. American Cancer Society, 1–7. https://doi.org/10.1002/9781118445112.stat06466.pub2

[15] Walter Kintsch. 1980. Learning from text, levels of comprehension, or: Why anyone would read a story anyway. *Poetics* 9, 1 (1980), 87 – 98. https://doi.org/10.1016/0304-422X(80)90013-3 Special Issue Story Comprehension.

[16] Peter Knees and Markus Schedl. 2016. *Introduction to Music Similarity and Retrieval*. Springer Berlin Heidelberg, Berlin, Heidelberg, 1–30. https://doi.org/10.1007/978-3-662-49722-7_1

[17] Peter Knees, Markus Schedl, and Masataka Goto. 2020. Intelligent User Interfaces for Music Discovery. *Transactions of the International Society for Music Information Retrieval* 3(1) (2020), 165–179.

[18] Volker B. Koettgen and Timm Seeger. 2020. Cafe: Coarse-to-Fine Neural Symbolic Reasoning for Explainable Recommendation. *CIKM 2020* (2020). https://doi.org/10.1145/3340531.3412038

[19] Shou De Lin and Hans Chalupsky. 2003. Unsupervised link discovery in multi-relational data via rarity analysis. *Proceedings - IEEE International Conference on Data Mining, ICDM* (2003), 171–178. https://doi.org/10.1109/icdm.2003.1250917

[20] Rishabh Mehrotra, Chirag Shah, and Benjamin Carterette. 2020. Investigating Listeners' Responses to Divergent Recommendations. In *Fourteenth ACM Conference on Recommender Systems* (Virtual Event, Brazil) *(RecSys '20)*. Association for Computing Machinery, New York, NY, USA, 692–696. https://doi.org/10.1145/3383313.3418482

[21] Daniel Müllensiefen, Bruno Gingras, Jason Musil, and Lauren Stewart. 2014. The musicality of non-musicians: An index for assessing musical sophistication in the general population. *PLoS ONE* 9, 2 (2014). https://doi.org/10.1371/journal.pone.0089642

[22] Cataldo Musto, Pierpaolo Basile, Pasquale Lops, Marco de Gemmis, and Giovanni Semeraro. 2017. Introducing linked open data in graph-based recommender systems. *Information Processing & Management* 53, 2 (2017), 405 – 435. https://doi.org/10.1016/j.ipm.2016.12.003

[23] Cataldo Musto, Pierpaolo Basile, and Giovanni Semeraro. 2019. Embedding Knowledge Graphs for Semantics-Aware Recommendations Based on DBpedia. In *Adjunct Publication of the 27th Conference on User Modeling, Adaptation and Personalization* (Larnaca, Cyprus) *(UMAP'19 Adjunct)*. Association for Computing Machinery, New York, NY, USA, 27–31. https://doi.org/10.1145/3314183.3324976

[24] Cataldo Musto, Tiziano Franza, Giovanni Semeraro, Marco de Gemmis, and Pasquale Lops. 2018. Deep Content-Based Recommender Systems Exploiting Recurrent Neural Networks and Linked Open Data. In *Adjunct Publication of the 26th Conference on User Modeling, Adaptation and Personalization* (Singapore, Singapore) *(UMAP '18)*. Association for Computing Machinery, New York, NY, USA, 239–244. https://doi.org/10.1145/3213586.3225230

[25] Cataldo Musto, Fedelucio Narducci, Pasquale Lops, Marco De Gemmis, and Giovanni Semeraro. 2016. ExpLOD: A framework for explaining recommendations based on the linked open data cloud. *RecSys 2016 - Proceedings of the 10th ACM Conference on Recommender Systems* (2016), 151–154. https://doi.org/10.1145/2959100.2959173

[26] Cataldo Musto, Fedelucio Narducci, Pasquale Lops, Marco de Gemmis, and Giovanni Semeraro. 2019. Linked open data-based explanations for transparent recommender systems. *International Journal of Human Computer Studies* 121 (2019), 93–107. https://doi.org/10.1016/j.ijhcs.2018.03.003

[27] Ingrid Nunes and Dietmar Jannach. 2017. A systematic review and taxonomy of explanations in decision support and recommender systems. *User Modeling and User-Adapted Interaction* 27, 3-5 (2017), 393–444. https://doi.org/10.1007/s11257-017-9195-0 arXiv:2006.08672

[28] Vito Claudio Ostuni, Tommaso Di Noia, Roberto Mirizzi, and Eugenio Di Sciascio. 2014. A Linked Data Recommender System Using a Neighborhood-Based Graph Kernel. In *E-Commerce and Web Technologies*, Martin Hepp and Yigal Hoffner (Eds.). Springer International Publishing, Cham, 89–100.

[29] E. Pampalk and M. Goto. 2007. MusicSun: A New Approach to Artist Recommendation. In *in Proc. of the ISMIR Intl. Conf. on Music Information Retrieval*. 205–210.

[30] Alexis Papadimitriou, Panagiotis Symeonidis, and Yannis Manolopoulos. 2012. A generalized taxonomy of explanations styles for traditional and social recommender systems. *Data Mining and Knowledge Discovery* 24 (05 2012), 555–583. https://doi.org/10.1007/s10618-011-0215-0

[31] Alexandre Passant. 2010. dbrec - Music Recommendations Using DBpedia. *The Semantic Web – ISWC 2010* 1380 (2010), 1–16.

[32] T. Pohle, P. Knees, M. Schedl, E. Pampalk, and G. Widmer. 2007. "Reinventing the Wheel": A Novel Approach to Music Player Interfaces. *IEEE Transactions on Multimedia* 9, 3 (2007), 567–575. https://doi.org/10.1109/TMM.2006.887991

[33] M. Porter. 1980. An algorithm for suffix stripping. *Program* 14 (1980), 130–137.

[34] Cartic Ramakrishnan, William H. Milnor, Matthew Perry, and Amit P. Sheth. 2005. Discovering informative connection subgraphs in multi-relational graphs. *ACM SIGKDD Explorations Newsletter* 7, 2 (2005), 56–63. https://doi.org/10.1145/1117454.1117462

[35] Roger C. Schank. 1979. Interestingness: Controlling inferences. *Artificial Intelligence* 12, 3 (1979), 273 – 297. https://doi.org/10.1016/0004-3702(79)90009-2

[36] Nava Tintarev and Judith Masthoff. 2011. *Designing and Evaluating Explanations for Recommender Systems*. Springer US, Boston, MA, 479–510. https://doi.org/10.1007/978-0-387-85820-3_15

[37] Kosetsu Tsukuda and Masataka Goto. 2020. Explainable Recommendation for Repeat Consumption. In *Fourteenth ACM Conference on Recommender Systems* (Virtual Event, Brazil) *(RecSys '20)*. Association for Computing Machinery, New York, NY, USA, 462–467. https://doi.org/10.1145/3383313.3412230

[38] G. Tzanetakis. 2001. Automatic Musical Genre Classification of Audio Signals. In *in Proc. of the ISMIR Intl. Conf. on Music Information Retrieval*. 205–210.

[39] Jesse Vig, Shilad Sen, and John Riedl. 2009. Tagsplanations: Explaining Recommendations Using Tags. In *Proceedings of the 14th International Conference on Intelligent User Interfaces* (Sanibel Island, Florida, USA) *(IUI '09)*. Association for Computing Machinery, New York, NY, USA, 47–56. https://doi.org/10.1145/1502650.1502661

[40] Morgan K. Ward, Joseph K. Goodman, and Julie R. Irwin. 2014. The same old song: The power of familiarity in music choice. *Marketing Letters* 25, 1 (2014), 1–11. https://doi.org/10.1007/s11002-013-9238-1

[41] Kangzhi Zhao, Xiting Wang, Yuren Zhang, Li Zhao, Zheng Liu, Chunxiao Xing, and Xing Xie. 2020. Leveraging Demonstrations for Reinforcement Recommendation Reasoning over Knowledge Graphs. *SIGIR 2020 - Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval* (2020), 239–248. https://doi.org/10.1145/3397271.3401171