Inferring Descriptions and Similarity for Music from Community Metadata

Brian Whitman[1], Steve Lawrence[2]
[1] MIT Media Lab, Music, Mind & Machine Group, 20 Ames St., E15-491, Cambridge, MA 02139
[2] NEC Research Institute, 4 Independence Way, Princeton, NJ 08540
bwhitman@media.mit.edu, lawrence@necmail.com

Abstract

We propose methods for unsupervised learning of text profiles for music from unstructured text obtained from the web. The profiles can be used for classification, recommendation, and understanding, and may be used in conjunction with existing methods such as audio analysis and collaborative filtering to improve performance. A formal method for analyzing the quality of the learned profiles is given, and results indicate that they perform well when used to find similar artists.

1 Introduction

Music retrieval and recommendation systems are becoming increasingly common as the computational resources required to handle digital audio are becoming more widespread. Current music recommendation systems typically use collaborative filtering or audio content-based features for recommendation. In collaborative filtering (Resnick, Iacovou, Suchak, Bergstrom, and Riedl 1994; Pennock, Horvitz, Lawrence, and Giles 2000), items are recommended based on the interests of other users that share interests with the current user. For audio content-based recommendation, similarity between songs or artists is computed based on audio analysis (e.g., based on FFT or wavelet analysis). Both methods have drawbacks, for example collaborative filtering may create a feedback loop with popular artists leading to a low probability of recommending new or (currently) unpopular artists. Audio content-based recommendation is difficult because the current state-of-the-art is unable to extract many high level features of interest with high accuracy.

In addition to the advances in information access created by the web, the web also represents an increasingly large fraction of human interests in machine processable form. In particular, the web contains an increasingly large amount of music-related information, with many web sites and discussion groups devoted to music.

We propose a model for music description and similarity based on analysis of the text contained in web pages, discussion groups, or other sources. We can take advantage of the continuous updates to information available on the web in order to create a dynamic representation that is updated frequently, taking into account the "buzz factor" for a particular artists or song, for example.

The text-based representations that we learn, which we call *community metadata*, can be used for understanding or similarity computation, in query-by-description systems, or in conjunction with existing collaborative and audio content-based systems in order to improve performance.

2 Background

Much of this work is a combination of techniques that have proved to be successful for information retrieval applied to the music domain. Music similarity has both mathematical and cognitive (Hofmann-Engl 2001) underpinnings. Understanding a piece of music enough to characterize genre or even artist (Whitman, Flake, and Lawrence 2001; Berenzweig, Ellis, and Lawrence 2002) is a common problem usually attacked by studying the spectral characteristics of the audio. In (Yang 2001), attempts are made to abstract the content from the style in a manner that could recognize "cover versions" of songs already in a database.

The text approaches used in this paper stem from previous work in natural language processing for information retrieval. For example, in (Evans and Zhai 1996) extracted noun phrases are used to aid a query task. For an overview of noun phrases and their grammar, see (Evans and Klavans 2000).

Although more specific than our application, crawling the web for music information retrieval was studied in (Cohen and Fan 2000), where users' "favorite artist lists" were classified and parsed autonomously to aid in recommendation.

3 Architecture

Our system works by querying web search engines for pages related to artists, downloading the pages, ex-

tracting text and natural language features, and analyzing the features to produce textual summary descriptions of each artist. These descriptions are then used to compute similarity between artists.

3.1 Artists

Our analysis uses a set of about 400 artists, which were the most popular artists appearing on OpenNap, a popular Napster-alternative sharing service, during a three week period in August, 2001. A software agent retrieved 1.6 million user – song entries (user x has song y in their shared folder), which we use later in this paper as user preference data. We did not download any song files from OpenNap. The top 1,000 albums from this set, chosen for maximal song coverage, were purchased and encoded onto a disk server. Related work analyzes the audio content of this database (Whitman, Flake, and Lawrence 2001; Berenzweig, Ellis, and Lawrence 2002).

Note that only one out of every four filenames our agent collected were mapped to an actual song name from a list of 700,000 current songs from All Music Guide (`www.allmusic.com`), due to typos and under-described data. This problem plagues similar systems (Pachet and Laigre 2001).

3.2 Similarity Data

We obtained artist similarity data from the All Music Guide, which typically lists three to five similar artists for each artist (more for popular artists), which we believe are manually selected by All Music editors. We use this manually created similarity data as a "ground truth" for analysis.

This similarity data has a number of limitations – for example, artist similarity is subjective, the artists listed are subject to the knowledge and preferences of the editors involved, the degree of similarity is not provided, different editors may use different criteria for selecting similar artists, and often only a small number of similar artists is provided. For these reasons, we do not expect our system to reproduce the All Music similarity lists, however we do expect relative comparisons to be useful, where different systems are compared to the All Music data.

3.3 N-grams, Part-of-Speech Tagging, and Noun Phrase Extraction from Freeform Text

Our input feature space to the system comes from Klepmit, a natural language feature extractor we developed for freeform web-extracted text. Klepmit takes as input a query term (artist name) which we augment with the search terms "music" and "review." The "review" search enhancement serves to limit the results to topical text about the artist (hopefully a review of an album, song, or concert.) Many results for the single-term only query "Madonna," for example, return splash pages or marketing concerns. The "music" search enhancement similarly hopes to limit common-word artist names such as "War" or "Texas" to return only musically-related pages.

We send the query to a search engine and then download up to 50 of the top returned pages. Each page is fed to a HTML parser that extracts the screen-viewable text. (The parser renders the page to disk instead of the screen, removing all images). We then remove all extraneous whitespaces and special characters and begin the process of feature extraction. We extract n-grams (sequences of ordered words having n words) for $n = 1$ (n1 or unigrams) and $n = 2$ (n2 or bigrams) from each page. We also feed the plain text input to a part-of-speech tagger [Brill's (Brill 1992)], which fits each single word into a part of speech class (noun, verb, pronoun, adjective, etc.). Finally, we apply a noun phrase (NP) chunker [Penn's baseNP (Ramshaw and Marcus 1995)].

3.4 Noun Phrases in Information Retrieval

Noun phrases can be thought of as a noun extended with a maximal amount of descriptive text surrounding it. There is a defined grammar for noun phrase extraction, and once part-of-speech tagging has occurred, a simple rule-based NP chunker can operate on any amount of text. Noun phrases suggest more than a simple bi- or tri-gram since their content is limited to one "idea." In the music domain, the sentence "Metallica employs screeching heavy metal guitars" leads to both "metal guitars" and "screeching heavy metal guitars" as noun phrases, but only the first is a possible bigram. Noun phrases can also serve as a simple noise reduction technique. A possible trigram from the above text could be "employs screeching heavy," which on its own does not provide much in the way of semantic description. But the NP extractor would retrieve the maximal NPs "Metallica" and "screeching heavy metal guitars".

The intuitive descriptive nature of noun phrases leads us to believe that they should perform better than n-grams in the same retrieval or description task.

3.5 Artist Term Extraction

An important part of our feature space is the "artist term" set. We parse the 1-gram list for terms that appear in the list of the top 6,000 artists found in our peer-to-peer crawling. By doing this, we hope to be able to designate a section of our feature space to "similar artist" explanations. Many reviews of artists use other similar artists as touchstones to describe the music, and by creating a feature space that directly makes use of this, we may gain greater accuracy in our evaluation.

3.6 Adjective Term Extraction

Our intuition led us to choose an adjectives-only subset of the n1 class as a semantically descriptive feature set. The adjectives term set consists of every n1 term tagged as an adjective by the part of speech tagger. The adjectives encapsulate a large amount of generalized descriptive content concerning the artists.

There are two important distinctions between the adjective term space and the other sets:

- The adjective set is human-readable and understandable. For the entire list of unigrams, important descriptive terms tend to get lost among common words, technical terms, Internet-specific terms and typos. While we describe below viable methods for extracting the most generalizable set of these terms, the adjective set is immediately recognizable and readable due to the extra layer of simple language processing which functions as a noise reduction technique. For applications such as query-by-description and description synthesis, the adjectives set is very useful.

- The adjective set is orders of magnitude smaller than the rest. The identified adjectives compose only about 1% of the unigrams found from our web crawls. An average adjective set for an artist is only 100 terms. The smaller number of terms helps speed learning and reduce complexity.

Because of these distinctions, we find that a different scoring metric for weighting adjective terms is necessary, which we describe below.

3.7 Evaluation Stage

After extracting the features, we compute term frequency and document frequency for each term type in each artist set. Term frequency (f_t) was defined as the percentage of retrieved pages that contained the given term (treating each retrieved page separately). Document frequency (f_d) was computed across the entire retrieved set, treating each artist as a document. We treat both f_t and f_d as a normalized probability between 0 and 1 for the entire artist space, and compute the TF-IDF [Term Frequency x Inverse Document Frequency (Salton and McGill 1983)] value of each term, which we also normalize between the local minimum and maximum values for each artist.

To evaluate our feature space, we investigate how well the system can predict the edited list of similar artists described earlier. The computation is based on "term overlap." If two artists share a term in their feature space, we say that those terms overlap with an associated *overlap score*. The scores for overlap are accumulated to create a numerical similarity metric between two artists. We compute overlap for all term types that we have extracted. For each artist we compute overlap with every artist in the known-edited similarity set (from All Music). We then average the overlap scores and compare them to the average overlap between the artist in question and the same number of randomly chosen artists. Using this method we obtain two evaluation metrics: a per-artist accuracy score (how well can we predict similar artists versus random artists using our textual feature space?), and an average overlap enhancement score (on average, how much more overlap within the feature space do similar artists have compared to randomly chosen artists?) The overlap improvement is considered in the average only when the similarity list is accurately predicted.

Due to the binary present or not-present nature of the edited ground truth similarity provided by All Music, we concentrate more on the second metric as overall fitness for our feature space. There are many instances in which the lists can not be considered complete or up-to-date. Therefore, we use our ground truth metric as a guide, and not a requirement for artist similarity.

We also discuss below another artist similarity metric created using peer-to-peer preference data that performs just as well and has the benefit of providing a continuous measure.

To compute the score of two terms having overlap, we experimented with various thresholding and smoothing metrics. The score of an overlap could simply be 1 (a match of a term on two artists' pages) or it could a function of the term and/or document frequency. In the former case, common words such as "music" or "album" get very high overlap among all artists, and typically do not retrieve musically intelligent terms. Considering this, we use a metric that is based on the TF-IDF value of the term in question: $\frac{f_t}{f_d}$. TF-IDF measures topical importance relating to a term by computing the term frequency (how often is appears relating to a topic) vs. document frequency (down-weighting by the amount of times it appears in general.) For example, the term "music" or "rock" might have a high f_t but also a high f_d, thus a low TF-IDF score. But, for Metallica, "Hetfield" (a band member's name) would have a high f_t and a very low f_d, causing the term to rank high. However, limiting scoring to high TF-IDF values will only try to match very specific terms such as: band members' last names, song titles, etc. In our experiments we investigate down-weighting very rare terms in addition to down-weighting very common terms.

The intended goal during the evaluation is to show that the extracted feature space can be valuable for computing musical similarity, and to verify the fitness of the representation. However, the space created by Klepmit can be used for many different tasks, which we discuss below.

n1 Term	Score	n2 Term	Score	np Term	Score	adj Term	Score	art Term	Score
voulez	0.0567	dancing queen	0.0707	dancing queen	0.0875	perky	0.8157	priscilla	0.0494
bjorn	0.0556	mamma mia	0.0622	mamma mia	0.0553	nonviolent	0.7178	burzum	0.0180
priscilla	0.0494	disco era	0.0346	benny	0.0399	swedish	0.2991	amorphis	0.0172
andersson	0.0446	winner takes	0.0307	chess	0.0390	international	0.2010	keaggy	0.0145
chiquitita	0.0424	chance on	0.0297	its chorus	0.0389	inner	0.1776	crabs	0.0140
muriel	0.0370	swedish pop	0.0296	vous	0.0382	consistent	0.1508	vous	0.0136
swedes	0.0359	my my	0.0290	the invitations	0.0377	bitter	0.0871	basia	0.0131
frida	0.0320	s enduring	0.0287	voulez	0.0377	classified	0.0735	mahalia	0.0121
sera	0.0231	and gimme	0.0280	something's	0.0374	junior	0.0664	connors	0.0115
collette	0.0231	enduring appeal	0.0280	priscilla	0.0369	produced	0.0616	placido	0.0097

Table 1: Top 10 terms of each type for ABBA. The score is TF-IDF for adj (adjective), and Gaussian weighted TF-IDF (see Section 5.1) for term types n1 (unigrams), n2 (bigrams), np (noun phrases) and art (artist). Self references (terms including "ABBA") were removed. Note that each term type could have a specific use: adjectives for description, n1 for classification, etc.

n1 Term	Score	n2 Term	Score	np Term	Score	adj Term	Score	art Term	Score
gibbons	0.0774	beth gibbons	0.1310	beth gibbons	0.1648	cynical	0.2997	gibbons	0.0774
dummy	0.0576	sour times	0.0954	trip hop	0.1581	produced	0.1143	rasputina	0.0411
displeasure	0.0498	blue lines	0.0718	dummy	0.1153	smooth	0.0792	latimer	0.0325
nader	0.0490	17 feb	0.0675	goosebumps	0.0756	dark	0.0583	aeroplanes	0.0321
tablets	0.0479	lumped into	0.0665	soulful melodies	0.0608	particular	0.0571	towa	0.0315
godrich	0.0479	which come	0.0635	rounder records	0.0499	loud	0.0558	tei	0.0315
irks	0.0467	mellow sound	0.0573	dante	0.0499	amazing	0.0457	retsin	0.0277
corvair	0.0465	in together	0.0519	may 1997	0.0499	vocal	0.0391	woob	0.0270
durban	0.0461	musicians will	0.0494	sbk	0.0499	unique	0.0362	richter	0.0269
farfisa	0.0459	enough like	0.0494	grace	0.0499	simple	0.0354	spacemen	0.0235

Table 2: Top 10 terms for Portishead. See Table 1. Here, the noun phrase and adjective terms seem to give the best descriptions.

Term	TF-IDF
perky	0.8157
nonviolent	0.7178
swedish	0.2991
international	0.2010
inner	0.1776
consistent	0.1508
bitter	0.0871
classified	0.0735
junior	0.0664
produced	0.0616
romantic	0.0607
raw	0.0520

Table 3: An example of Gaussian smoothing on the adjectives term set for ABBA, showing that the generalizing terms are amplified while the too-specific or too-general terms are attenuated. The more important the Gaussian function thinks a term is, the darker the shading.

4 Peer-to-Peer Similarity

We investigated other methods of artist similarity besides the text overlap method. The user preference data mentioned above was used to create a similarity measure of artists based completely on user collections. We defined a collection as the set of artists a user had songs by on their shared folder during the OpenNap crawl. If two artists frequently occur together in user collections, we consider them similar via this measure of community metadata. We also define a collection count $\mathcal{C}(artist)$ which equals the number of users that have $artist$ in their set. $\mathcal{C}(a, b)$, likewise, is the number of users that have both artists a and b in their set.

However, one particular problem of this method is that extremely popular artists (such as Madonna) occur in a large percentage of users' collections, which down-weights similarity between lesser-known artists. We developed a scoring metric that attempts to alleviate this problem. Given two artists a and b, where a is more popular than b (i.e., $\mathcal{C}(a) \geq \mathcal{C}(b)$), and a third artist c which is the most popular artist in the set; they are considered similar with normalized weight:

$$\mathcal{S}(a, b) = \frac{\mathcal{C}(a, b)}{\mathcal{C}(b)}(1 - \frac{|\mathcal{C}(a) - \mathcal{C}(b)|}{\mathcal{C}(c)}) \qquad (1)$$

The second term is a "popularity cost" which down-weights relationships of artists in which one is very popular and the other is very rare.

Since All Music Guide's average count of similar artists is five, we compute a matrix $\mathcal{S}(a, b)$ among every pair of artists and sort the top five for each one. We note that outside of the top five metric, the similar-

All Music Guide	OpenNap	n2 Overlap
Erasure	Culture Beat	Pink
Madonna	Thompson Twins	New Order
New Order	New Order	Duran Duran
Magnetic Fields	Blondie	KC & The Sunshine Band
Alphaville	Erasure	Culture Club
	Duran Duran	Procol Harum
	Roxette	Placebo
	Eurythmics	Westlife
	Ace of Base	Rage Against the Machine
	Wham	Alphaville
	Depeche Mode	La Bouche
	A-Ha	Samantha Mumba

Table 4: Top similar artists for "The Pet Shop Boys" using three different metrics. AMG only provides five artists (that were in our set of 414), while the other metrics are continuous (here we limit it to the top 12).

ity metric $\mathcal{S}$ creates a continuous measure of similarity much like our term overlap metrics.

5 Experiments and Results

Our experiments concentrate on evaluating the fitness of our representation by comparing the performance in computing artist similarity with an edited collection. We note that our representation is suitable for many tasks; but artist similarity is well-posed and we can perform formal evaluation with "ground truth" data.

During each of the following experiments, we ran a system that computes overlap of terms. Our grounding assumption is that similar artists share features in our space, and that our representation allows for enough generality to classify artists into similar clusters. To evaluate, we compare the performance of our varying feature types in the task of predicting the All Music Guide's similarity lists (for each of our 414 artists, AMG on average lists 5 other artists also in our set that are known similar).

For each artist in our set, we take the top n terms from their feature space. n is defined as a rough minimum for the size of the feature space; we want each artist to have the same amount of terms for comparison purposes. For the n1 term type, for example, n is 1000 (n2: n=5000, np: n=5000, adjectives: n=50, artist: n=500). The top n terms are sorted by the overlap scoring metric (see below). We then compare this feature space against every artist in the current artists' edited similarity list. The overlap scoring metric is averaged for each similar artist. We then do the same for a randomly chosen set of artists. If the overlap score is higher for the similar artist set, we consider that our feature space correctly identified similar artists. The percentages shown below indicate the percentage of artists whose similar cluster was predicted. We expect this task to be relatively easy, i.e., we expect percentages $\gg 50\%$. Note although that the entire set of artists (which correlates with the interests of OpenNap users) is predominately rock and pop with few artists from other styles of music.

We also compute a more powerful metric which we call *overlap improvement*, which is the ratio between overlap scores for similar artists compared to randomly chosen artists. A higher overlap improvement indicates a stronger confidence of the feature space for this task.

5.1 Overlap Scoring Metrics

We use two different overlap scoring metrics in our experiments, each of which is suited for different term types. The nature and size of the n1, n2 and np sets (in the tens of thousands for each artist) led us to believe that we needed a way to emphasize the terms found in the middle of the span of IDF values. The intuition is that very rare words, such as typos and off-topic words rarely used on music pages, should be down-weighted in addition to very common words such as "the". To achieve this, we used a Gaussian smoothing function that, when given appropriate μ and σ (mean and standard deviation) values, can down-weight both very common and very rare terms:

$$\frac{f_t e^{-(log(f_d)-\mu)^2}}{2\sigma^2} \quad (2)$$

where f_d is renormalized such that the maximum is the total document count of 20,700 (50 pages each from 414 artists.) To compute the overlap score, we simply add the Gaussian-weighted result for each term found in both the comparison and the base artists' sets. μ and σ were chosen with an intuition on the size and nature of each set. We also experimented with varying values. For almost all term sets (n1, n2, np and artist), we used a μ of 6 and a σ of 0.9.

For the adjective set, however, the filtering of non-adjectives already eliminates typos and many non-music-related specific terms, so we expect the standard TF-IDF weighting to perform well.

For both metrics, we only consider the first 25 matches in the cumulative score in order to reduce the computational load.

5.2 TF-IDF Scoring Metric

We computed the overlap improvement and per-artist accuracy using the standard TF-IDF scoring metric (without the Gaussian weighting), as shown in Table 5.

	n1	n2	np	adj	art
Accuracy	78%	80%	82%	69%	79%
Improvement	7.0x	7.7x	5.2x	6.8x	6.9x

Table 5: Results for the TF-IDF scoring metric for artist similarity. *Accuracy* is the percentage of artists for which the system correctly predicts the known-similar artists from the random artists. *Improvement* is the factor by which the overlap scores improve on average when the known-similar artists are correctly predicted.

The results appear good. From automatically crawled free-form text, the n1 term set can identify similar artists 78% of the time, and when it does, it has a relatively high confidence (the 7.0 times improvement in the overlap scores.) The longer n2 and np terms do slightly better on the accuracy metric, while the adjective set performs relatively poorly.

5.3 Gaussian Scoring Metric Results

With the Gaussian weighting in place, we then computed the overlap improvement and per-artist accuracy for each term type, as shown in Table 6.

	n1	n2	np	adj	art
Accuracy	83%	88%	85%	63%	79%
Improvement	3.4x	2.7x	3.0x	4.8x	8.2x

Table 6: Results for the Gaussian scoring metric for artist similarity.

As expected, accuracy improves for the n1, n2, and np sets compared to the standard TF-IDF metric, however the overlap improvement is lower. Our intuition behind the the improved accuracy but decreased overlap improvement is that the deemphasized rare terms are often misleading, resulting in higher accuracy, but these rare terms sometimes work very well, resulting in very high overlap for a fraction of artists and a higher average overlap improvement when not using the Gaussian weighting.

The accuracy for the adjective set is lower, in line with our previously mentioned expectations.

Sample terms for each case can be seen in Tables 1 and 2. Although the n1 and n2 terms sets often do not rank clearly understandable terms highly, the total term sets still perform well according to our artist similarity metrics. For understanding and query purposes, we note that the np and adjective types provide the most semantic content. Because of the extra steps taken in extracting them (both required a rule-based part of speech tagger with knowledge of the English language), we have more faith in their ability to succinctly describe an artist. The most general descriptive terms tend to come from the adjective set, which seems to strike a good balance between statistical significance and semantic fitness.

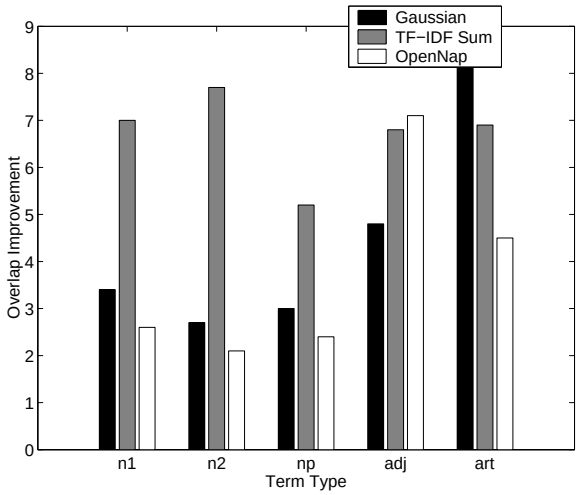

Figure 1: Comparison of overlap improvement scores across term types and scoring metrics.

5.4 Fitness of the Similarity Space

Since artist similarity is not a well-defined space (any list would have to be incomplete), we tried to compare our feature space against the more collaborative artist similarity metric computed from the OpenNap preference data (see Section 4). We limited each list of similar artists to the top 5 and used the same system to compute the term overlap against this new list. We chose the best of breed accuracy scoring metric per term: for n1, n2, np and artist term types this was the Gaussian overlap. For adjective term types we kept the TF-IDF sum overlap metric.

	n1	n2	np	adj	art
Accuracy	80%	82%	84%	68%	72%
Improvement	2.6x	2.1x	2.4x	7.1x	4.5x

Table 7: Similarity accuracy using OpenNap community data as ground truth.

We note that most of the results are similarly good. Figure 3 graphically compares the OpenNap similarity metric with the All Music Guide similarity lists.

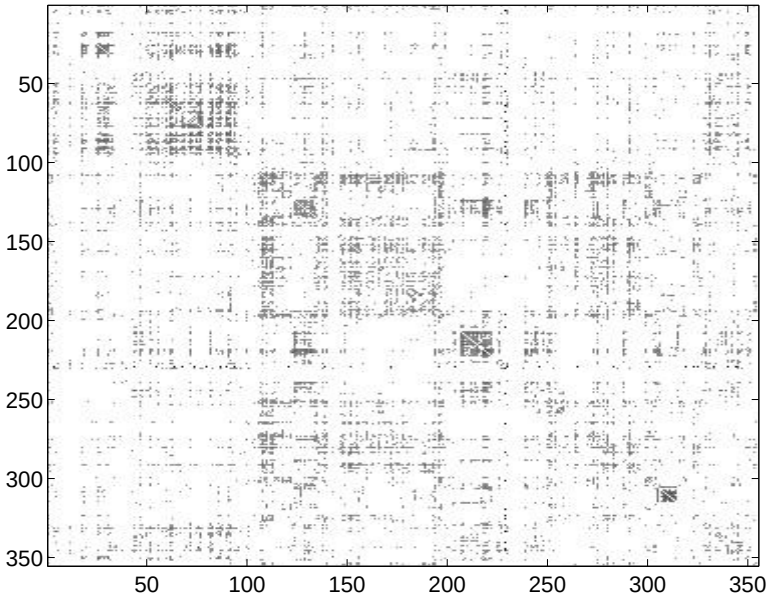

Figure 3: OpenNap similarity matrix. Dark dots show high artist to artist similarity. We arranged the artist list (in order on each axis) in a directed walk starting from Aerosmith. (We only see 354 artists due to the path-finder not being able to fit the remaining 60 artists in a similarity context.) Each adjacent artist is similar to each other (according to All Music Guide). We maximized global clustering along genres as well, by not moving between disconnected artist clusters until every path among similar artists was traversed. The intended effect is to show activity along the diagonal: varying sized clusters of boxes indicating that community metadata derived similarity, as defined by the OpenNap collaborative metric, is tightly aligned with the edited similarity set. There are deviations (the sparse activity outside the diagonal), which could be new relations that All Music Guide's editors have not yet entered.

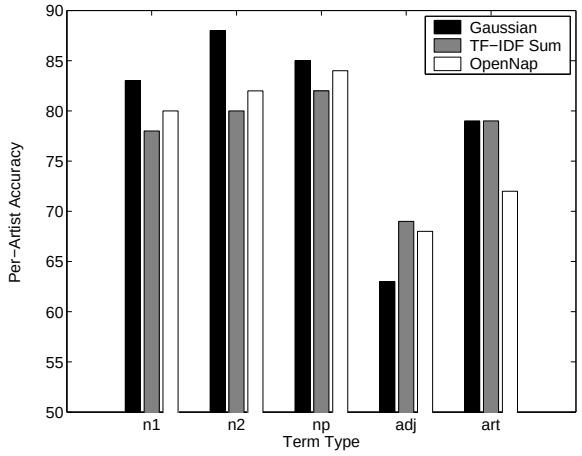

Figure 2: Comparison of per-artist accuracy scores across term types and scoring metrics.

6 Discussion and Future Work

Any application of this feature space could benefit from a combination of the varying term types, depending on the task and the amount of user involvement. A query-by-description system currently underway at MIT, for example, is using a subset of the adjective, noun phrase, artist and n1 term types. The generality of the adjective terms (like "loud" or "electronic") works quite well for describing large clusters of music, but the more specific user queries ("Something like Metallica, but quiet") take advantage of the respectively specific term types.

We note that perfect results for either of the fitness metrics (100% for per-artist accuracy, or a very high confidence) is unrealistic. We have uncertainties in the edited artist similarity list, as well as the crawled data. Some of the more popular artists in our collection have very poor retrieved terms: we attribute this to a collective Internet assumption that *everyone* knows about Madonna, for example. Even the query enhancements we applied did little to extract content directly concerning the music of some of the artists. Related to this, some of the artists with single term names with alternative meanings ("Texas," "Cure") retrieve thousands of unrelated documents.

To counteract this dilemma, we are developing community based crawlers that dynamically modify queries to include musically-salient terms extracted from this collective feature space. Since most of the artists do in fact return musically descriptive terms (especially in the adjectives term set), we can choose subsets of random artists to serve as filters during the crawling stage. A simple "musically relevant" webpage classifier has

already been built using this method augmented with a machine learning classifier.

7 Conclusions

We show that an application of community metadata performs an artist similarity task with good results compared to a human edited list. The combined power of these term-based representations can fill a gap in current music retrieval: understanding the "semantic profile" of an artist through a feature space that maximizes generality and descriptiveness. The collaborative source of the feature space has an important dynamic property, as well: as times and aesthetics change, so do people's perceptions of music. This representation can fully takes advantages of these facts, and allows for time-aware retrieval, understanding, and recommendation of music.

Acknowledgments

The authors wish to thank Adam Berenzweig, Deb Roy, Paris Smaragdis, and Barry Vercoe for their helpful discussions.

References

Berenzweig, A., D. Ellis, and S. Lawrence (2002). Using voice segments to improve artist classification of music. submitted.

Brill, E. (1992). A simple rule-based part-of-speech tagger. In *Proceedings of ANLP-92, 3rd Conference on Applied Natural Language Processing*, Trento, IT, pp. 152–155.

Cohen, W. W. and W. Fan (2000). Web-collaborative filtering: recommending music by crawling the web. *WWW9 / Computer Networks 33*(1-6), 685–698.

Evans, D. and J. Klavans (2000). Document processing with linkit. In *RIAO 2000*.

Evans, D. A. and C. Zhai (1996). Noun-phrase analysis in unrestricted text for information retrieval. In *Meeting of the Association for Computational Linguistics*, pp. 17–24.

Hofmann-Engl, L. (2001). Towards a cognitive model of melodic similarity. In *Proceedings of the 2nd Annual International Symposium on Music Information Retrieval*, Bloomington, Indiana, pp. 143–151.

Pachet, F. and D. Laigre (2001). A naturalist approach to music file name analysis. In *Proceedings of the 2nd Annual International Symposium on Music Information Retrieval*, Bloomington, Indiana, pp. 51–58.

Pennock, D., E. Horvitz, S. Lawrence, and C. L. Giles (2000). Collaborative filtering by personality diagnosis: A hybrid memory- and model-based approach. In *Proceedings of the 16th Conference on Uncertainty in Artificial Intelligence, UAI 2000*, Stanford, CA, pp. 473–480.

Ramshaw, L. and M. Marcus (1995). Text chunking using transformation-based learning. In D. Yarovsky and K. Church (Eds.), *Proceedings of the Third Workshop on Very Large Corpora*, Somerset, New Jersey, pp. 82–94. Association for Computational Linguistics.

Resnick, P., N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl (1994). Grouplens: An open architecture for collaborative filtering of netnews. In *Proceedings of ACM CSCW'94 Conference on Computer-Supported Cooperative Work*, pp. 175–186.

Salton, G. and M. McGill (1983). *Introduction to Modern Information Retrieval*. McGraw Hill.

Whitman, B., G. Flake, and S. Lawrence (2001, September 10–12). Artist detection in music with minnowmatch. In *Proceedings of the 2001 IEEE Workshop on Neural Networks for Signal Processing*, pp. 559–568. Falmouth, Massachusetts.

Yang, C. (2001). Music database retrieval based on spectral similarity. In *Proceedings of the 2nd Annual International Symposium on Music Information Retrieval*, Bloomington, Indiana, pp. 37–38.