Song Genre and Artist Classification via Supervised Learning from Lyrics

Adam Sadovsky
Xing Chen
CS 224N Final Project

I. Introduction

Motivation

The amount of raw data available online has increased dramatically over the past few years; in order for us to maintain the usability of this data we must develop effective ways to efficiently and automatically organize it. For our CS 224N final project, we chose to develop a classifier that classifies songs into genres and/or artists based solely on their lyrics. We primarily focused on developing lyric-specific features that would allow a classifier to easily distinguish between songs from different genres.

Project Description and Implementation

We decided to develop a classifier that can distinguish between three different genres of songs: rap (hip-hop), rock, and country. To collect training and testing data for our classifier, we used a program called EvilLyrics [8] to download a set of full albums for each of these genres. Representative albums were selected using album popularity charts such as those available on Amazon.com [7], Billboard.com [6], and BBC Radio 1 [5]. We later also analyzed the effectiveness of our classifier at classifying lyrics by artist; EvilLyrics was again used to download lyrics of all albums for each artist in our dataset.

After generating datasets (approximately 150 songs per genre or artist) and developing features, we used Maximum Entropy (Maxent) and Support Vector Machine (SVM) classifiers to classify lyrics (these were selected based on their performance on similar classification problems). SVMs, in their simplest implementation, learn a separating hyperplane between different groups of points (e.g. lyrics). Because SVMs don't require complex tuning of parameters, exhibit a "great ability to generalize given a small training corpora," and are well-suited for "learning in high-dimensional spaces" [4], they are ideal for the song lyric classification problem. For the SVM classifier, we used the Weka machine learning package [9]. For the Maxent classifier, we modified the PA2 Maxent code to run on song lyrics. We also experimented with several other classifiers in Weka (e.g. K-means, K-nearest-neighbors, naïve Bayes) but did not report these results.

Additional Notes

1. Because we didn't have a large corpus of training data, we implemented n-fold cross-validation (n=10 by default) within the Maxent framework in order to reduce the variance of reported classification accuracy. In addition to artificially inflating our training set size, using cross-validation in the Maxent classifier also allowed us to directly compare classification accuracy with Weka's SVM (which also uses ten-fold cross-validation).

2. We used part-of-speech (PoS) tag frequencies as a feature for lyric classification. To measure PoS tag frequencies, we used the Stanford Log-linear Part-Of-Speech Tagger [10], which labels each string in a given input file with a PoS tag. Before using the Stanford PoS Tagger, we also tried to use the MontyLingua NLP package [11] (which includes a PoS tagger among other NLP utilities), but we were unable to incorporate this package into our code (incompatibility issues with Java 1.5).

3. We wrote several Perl scripts to help us parse output files and prepare input files; these are available in the scripts directory of the submission folder.
   - webToMaxentFormal.pl takes an EvilLyrics output file and formats it for the Maxent classifier.
   - parseKMeans.pl takes the output of the Weka K-means clustering algorithm and prints statistics on the genre composition (i.e. percentages) for each cluster and vice versa.
   - filterMaxentWeights.pl takes the final Maxent weights (which are outputted by the modified Maxent code) and generates a bag of words for Weka classifiers (including SVM). See "feature selection" section for more information on this.

4. Significant changes were made to both MaximumEntropyClassifierTester.java and BasicLabeledDatum.java; for example, in Maxent's extractFeature method we wrote code to output Weka-compatible input files so that we could run Weka's SVM classifier.

II. Feature Selection

Bag of Words

Song lyrics are usually relatively short in length and are constructed from a relatively limited vocabulary. Thus, the selection of words in a song becomes one of its most important characteristics. There are two types of words that are encompassed by this analysis- content and function. First, past studies on authorship attribution have shown that common function words such as ‘of’ can be an effective marker of author style. Holmes [2] suggests that they may characterize an author’s writing so effectively because they are not entirely under his or her control. Tao et. al. also use them in their study of musical artist style [1]. In our case, differentiating between styles of authorship in songs may also help us differentiate between genres, since writers in a single genre will probably have more similar styles than between genres. Content words, on the other hand, allow us to extract semantic meaning from the song. Words such as ‘life’ or ‘love’ can be strong indicators of the song’s topic. By using these words, we can compare tendencies between different genres to choose different topics.

Selection

To choose a relevant bag-of-words, we used a measure of importance for each word taken from the output of our maximum entropy classifier. We first run Maxent on our data using all words (unigrams) as features and output the final label weights for each unigram feature. These weights are indicators of how many “votes” each word adds to each label when it appears in a song. In other words, for a given (word, label) pair, a large positive weight means that the label is more likely given the word, and a large negative weight means the label is less likely given the word. Thus, to judge importance of a single word, we use the measure:

$$\text{importance} = \max \left\{ \begin{array}{c} |\ \text{weight}(\text{word},\text{label}_i) - \text{weight}(\text{word},\text{label}_j)\ | \\ \text{for all pairs of labels } i,j \end{array} \right.$$

We then chose a bag of words whose importance was greater than a cutoff, determined to limit the size of the bag to approximately less than 50 words. This choice of bag size was evaluated empirically and selected for its relative performance. We note that importance measurement is only a relative measurement used to compare between words, and has no meaning when considered by itself. See Table 1 on the next page for a sample bag of words.

Discussion

We note several interesting features about the bag of words that we found. The content word “baby” shows up a lot more in country than rap and rock, and slightly more in rap than rock. This indicates that country songs tend to talk about love and girls more often than rock and rap. Furthermore, this corresponds to other “love song” words, such as “she” and “her,” both of which have a strong appearance in country songs. Also, we found that words about time, such as “when,” “day,” and “days” are also correlated in their appearance in country over rock or rap. This may be due to country singers singing about past experiences more often than rock or rap. Another interesting comparison is the use of personal pronouns. We see that “I” appears more often in rap than rock, but “I’m” and “me” appears more often in rock than rap. This may be an indicator of song topic selection. The word “I” is used for actions or declarations (for example, “I win”), while the words “I’m”, or “I” as an object in the form of “me” are often used to talk about what is happening to one’s own person (for example, “I am sad”). Thus, the contrast between the different uses of the pronoun “I” may allude to the difference in topics between rap and rock music.

Word Endings

The endings of words can indicate things such as verb tense, meaningful suffixes, and slang use. All of these things combined contribute to the semantic meaning of a song as well as the writing style of a song. For instance, the appearance of the “-er” ending can indicate the presence of verbs in a song, while the suffix “-ive” can signify the presence of adjectives.

To choose a set of relevant endings, we once again used Maxent to select a bag of endings based on the same importance weight shown above. We note that some of the endings overlap with counts of short words; this should be corrected in future work. A table of endings and their importance is shown in Table 2.

Line Length

The length of a line in the song can indicate several things dealing with the rhythm and pattern of a song’s acoustics. For example, frequent line breaks can indicate a song which flows in a much more choppy way than long lines with many words. In the case of rap, a long line may signify a smooth, unbroken train of words, while short lines show a rap style with frequent stops and breaks. Furthermore, short lines may indicate a singing style which draws out the sound of each single word. We used the average line length of a song in our classifier.

Number of Lines

Similar to line length, the number of lines can also hint at the acoustics behind the song lyrics. One can view the number of lines as the inverse of the line length; more lines may indicate a style with frequent stops, while fewer may indicate longer lines or drawn out words. Furthermore, the number of lines is an indicator of overall song length as well as the number of verses in the song. One issue we had with this was the replacement of repeated song lyrics in our datasets by the tokens (Chorus 2x) or (Verse 2x), etc.

Punctuation

The use of punctuation marks can reveal both lexical and acoustic information. The use of things apostrophes can indicate the use of slang and shorthand, and can be seen as a marker for an artist’s style of speech. The use of periods, commas and other sentence delimiters however, indicate stops in the song. These can reveal the song’s pattern of rhythm.

We do a simple count of punctuation in our feature. However, we note that in the future it would be better to count apostrophes and periods, commas, etc separately. One issue with punctuation was that random punctuation would sometimes be inserted in the lyrics. Also, “…” will confuse the count.

Part-of-Speech

Past work has shown that part-of-speech statistics are a useful feature in analysis of authorship. As noted by Li et al., POS features often reflect the characteristics of the writing [1]. In our case, this applies to both artist style as well as song topic. As with function words, a writer’s use of different POS can be a subconscious decision determined by the writer’s own style. Thus, again we propose that similarity between artists in a single genre may allow POS style across a genre to be similar. Furthermore however, POS can also be an indicator of the type of content in a song. Frequent use of verbs reveals a song that is about action; perhaps a song that is more story-oriented. Adjectives on the other hand, may indicate a song that is more descriptive in purpose.

In order to extract POS information, we used a part-of-speech tagger (Stanford POS tagger) to tag the words in a song. We then combine the POS into a “bag,” and enter into the feature vector a normalized count of each POS. This gives us frequency information on the parts of speech used in a song.

Repetition (Across Song)

Repetition plays a large role in the musical character of a song. Different artist and music styles tend to use repetition of words, phrases or verses in varying amounts. Thus, we first look at repetition across the entire song to detect repeated words, phrases, lines or verses. To score repetition, we simply count the number of words that are repeated. Thus, a single word repeated 5 times will add a count of 4, while a 4-word phrase repeated once will also add a count of 4. We then normalize the score against the total number of words.

Repetition (Within a Line)

In addition to repetition in the entire song, we also want to point out repetitions within a line. Generally, these have a different meaning than repetitions across a song; instead of being a repeated phrase or verse, these can be single words sung in multiples. For example, Matchbox 20 sings “Baby, baby, baby when all our love is gone…” These types of repeats are an indicator of artist style, and thus could also distinguish between genre styles. Clearly, this type of repetition is distinct from the type above; thus, we keep a separate count for number of repetitions within a line, and normalize against the number of words in the song.

III. Genre Classification

General Results

We used our features to train a Maxent model and n Support Vector Machine on 527 songs split roughly evenly between three genres: rock, rap and country. We then performed a 10-fold cross validation using our two classifiers with all features. The following are the resulting accuracies.

Classifier w/ All Features	Accuracy (%)
Maxent	76.45
SVM	81.21

SVM Confusion Matrix

{country, rap, rock}

```
   a   b   c   <-- classified as
 157   4  32 |   a = country
   6 152   3 |   b = rap
  46   8 119 |   c = rock
```

We see that our classifier performed exceptionally well in classifying rap, which is expected since rap is such a distinctive type of music. Country and rock however, is a finer distinction, and the confusion between them is where most of the errors occur. Furthermore, we see that more rock songs are classified as country than vice-versa. This may be due to the larger variation in rock songs; popular contemporary rock tends to take on a greater number of forms in terms of its themes, styles and topics than country music. Thus, it is expected that rock is the most difficult genre to classify.

Feature Analysis

We then performed a more detailed analysis of each of our features to explore their performance characteristics. Table 3, graph 1, and graph 2 summarize our results from these experiments. We note that the features were implemented slightly differently in Maxent than in SVM, and thus may behave differently.

Bag of Words

As expected, the bag of words was the most important single feature for both classifiers, as shown by its high performance alone. This is expected because it holds the most lexical and semantic information in a single feature. Thus, with subsequent features, we also test how well they do when combined with bag-of-words.

One interesting note is that both classifiers are also robust enough to perform almost as well without the bag-of-words feature. This may be attributed to the strength of the word endings feature.

We note that the implementation of bag-of-words was different for Maxent and SVM. Maxent performed better when we added every word we saw as a unigram feature, while SVM performed better when we selectively chose a small number (~40) of words to count as features in its feature vector.

It would be interesting to split the bag of words into function words and content words. This would allow a distinction between the style and content of a song.

Number of Lines

When used alone, the number of lines turned out to be a fairly strong feature for such a simple heuristic. However, when we look at its confusion matrix, we note that it classifies all songs as either rap or country. It does a fairly good job on rap, but fails miserably in classifying rock songs. This signifies that the number of lines between country and rap are significantly and consistently different, while the variation in rock prevents it from being accurately classified using this feature. Such results make sense, since rap is closer in style to rock than country; thus, a feature that looks at song style would pick out the distinction.

<u>Confusion Matrix (Number of Lines Only)</u>

```
   a   b   c   <-- classified as
 181  12   0 |   a = country
  54 107   0 |   b = rap
 145  28   0 |   c = rock
```

When used with bag-of-words, the number of lines barely improved Maxent, but improved SVM by 0.75%. When taken out of all features however, it decreases accuracy by almost 1% for both classifiers

Words per Line

When used alone, words per line turned out to be an effective feature for Maxent, but not for SVM. This is most likely due to the implementation of the feature. In Maxent, we add a feature for each observed count of words per line. This allows the classifier to have a sort of histogram distribution of words per line over the songs. In SVM however, we condensed words-per-line to be an average; this turned out to be a weaker feature. Our motivation was to avoid data sparsity, since it’s possible for few lines to have exactly the same number of words. In the future, an improvement could be made to bucket the lines.

With bag-of-words, words per line improved SVM but decreased accuracy in Maxent. This may be because the way we score unigrams in Maxent already takes into account the number of words in the song; thus, words-per-line becomes a less useful feature.

When removed from all features however, words per line decreased accuracy more in Maxent. This implies that its combination with another feature, possible lines per song, is necessary for it to make a contribution to the accuracy.

Punctuation

Punctuation is an extremely weak feature when used alone, scoring almost equivalent to a random choosing of genres (37%). When combined with bag of words however, it improved both classifiers by ~1%. We note that it is probably too specific of a feature to be useful in separating many songs. When removed, it only decreased the accuracy by a small, probably insignificant amount. It would be interesting to try to split up types of punctuation instead of doing a net count.

PoS

Part of speech statistics performed surprisingly well when used alone, scoring a third-best 61% in both Maxent and SVM. This indicates that it is a strong enough indicator of style that it can make significant distinctions in the data. As the confusion matrix shows, rap has a very distinctive use of POS, while rock and country have more variation in their style. In general however, the classifiers perform well using this feature, placing the majority of each genre correctly.

Confusion Matrix (POS Only)

```
   a   b   c   <-- classified as
 108  36  49 |   a = country
  21 128  12 |   b = rap
  59  27  87 |   c = rock
```

When combined with Maxent however, the performance went down nearly 4%. We hypothesized that this may be due to a sparsity of data; if not enough POS are seen, an accurate evaluation cannot be made. Due to time restrictions, we were not able to obtain more lyrical data. However, to explore whether POS may be more helpful on a large data set, we also went back to observe its effect on categorization on companies, drugs, movies, persons and place titles from PA2. There, we found that the results improved when we added POS to a Maxent model. Thus, it is possible that more data will allow POS to perform better.

PA2 Maxent Model

Trigram only	= 83.02%
Trigram + POS	= 84.24%

POS helped SVM a significant amount (~3%) when combined with bag of words. It also hurt both classifiers by ~1% when removed from the full list of features.

Repeats (Across Song)

Word repeats across a song performed poorly (almost random) when used alone on Maxent, and only a little bit better on SVM. When combined with bag of words, it improved Maxent by ~1% and SVM by ~4%. Overall, SVM manages to use this feature more effectively. This may be due to the implementation; Maxent simply counts repeats, while in SVM we give a fraction repeated normalized against total number of words.

When removed from all features however, repeats barely affect SVM, and decrease Maxent by ~2.5%. Thus, repeats must work with some other feature in Maxent, perhaps number of lines or words per line, as both are measures of a song’s length. This feature’s performance would most likely be improved if we had data which was guaranteed to have all the words; things such as (Chorus 2x) will decrease its relevance.

Repeats (Within Line)

Word repeats within a line performed similarly poorly on its own. It did not affect Maxent accuracy when combined with bag of words, but improved SVM by ~1%. The same normalization issue may be taking place here, as Maxent’s counts are not normalized.

When removed from all features, repeats within a line decrease accuracy by ~0.5%. Thus, this feature was not very strong. It is possible that more data would improve it. Also, it has the same issues as repeating across the song.

Word Endings

Word endings performed nearly as well as bag-of-words on its own under both classifiers, which is surprising despite the overlap between endings and short words.

<u>Confusion Matrix (Endings Only)</u>

```
   a   b   c   <-- classified as
 145   6  42 |   a = country
  22 132   7 |   b = rap
  61   6 106 |   c = rock
```

We see that once again, the trend continues of classifying rap well and confusing country and rock. Also, here we see that rock tends to be classified as country more than vice versa, again showing its high variation.

When combined with bag of words however, we only saw a ~4% improvement in SVM, and a slight decrease in Maxent. This may be because bag of words and endings end up “noticing” many of the same characteristics of a song, such as content and function word usage style. This will occur even more because endings encompass many short function words.

When removed from the features, endings decrease accuracy by a significant amount (~3%). It would be interesting to try removing the overlap between endings and bag of words, and testing its performance.

IV. Classifying Artists

Description

After analyzing the results from genre classification, we decided it would be interesting to see if our classifier performs better when each group of lyrics is by the same artist (i.e. all rock lyrics by one artist, all country lyrics by another artist, etc.). In the genre datasets, we found (by manual inspection of the data) that different artists within the same genre often had rather different lyric styles. Especially given the relatively small size of our genre datasets, these differences could potentially make it hard for a classifier to learn general descriptors of a genre's lyrics.

We theorized that lyrics from a single artist (and, consequently, from the same genre) would tend to be more similar than lyrics of many different artists from the same genre. In addition, even artists from the same genre often times have distinct lyric styles. To quantitatively assess these hypotheses, we used EvilLyrics to generate five more datasets:

1.	The Beatles	rock	~13 albums
2.	U2	rock	~13 albums
3.	Blink-182	rock	~7 albums
4.	Garth Brooks	country	~12 albums
5.	Snoop Dogg	rap	~9 albums

From these artist sets we generated three combined datasets for classification: {beatles, u2, blink}, {snoop, beatles, garth}, and {beatles, u2, blink, snoop, garth}. Since The Beatles, U2, and Blink-182 are all rock bands, we were able to both test whether our classifier can distinguish different artists alone (i.e. when the artists all belong to the same genre), and also whether our genre-classifier improves when each genre's songs are all by the same artist.

Results

All accuracy measurements were made using ten-fold cross-validation over the entire dataset. The Weka package was used for SVM classification.

Dataset	Classifier	Accuracy
{beatles, u2, blink}	Maxent	0.70464
{beatles, u2, blink}	SVM	0.72050
{snoop, beatles, garth}	Maxent	0.83866
{snoop, beatles, garth}	SVM	0.84115
{beatles, u2, blink, snoop, garth}	Maxent	0.70145
{beatles, u2, blink, snoop, garth}	SVM	0.68229

SVM confusion matrices:

{beatles, u2, blink}

```
  a  b  c   <-- classified as
 98 17 15 |  a = beatles
 19 91  8 |  b = u2
 18 13 43 |  c = blink
```

{snoop, beatles, garth}

```
   a   b   c   <-- classified as
 104   3  23 |   a = beatles
   8 126   3 |   b = snoop
  21   3  93 |   c = garth
```

{beatles, u2, blink, snoop, garth}

```
   a   b   c   d   e   <-- classified as
  86  14  11   4  15 |   a = beatles
  17  75   9   1  16 |   b = u2
  13  10  36   2  13 |   c = blink
   4   3   0 125   5 |   d = snoop
  19  18   8   1  71 |   e = garth
```

Analysis

As expected, our genre classifier performed better when all songs from each genre were by the same artist (Maxent: 83.9% > 76.4%, SVM: 84.1% > 81.2%). As with variable-artist classification, almost all rap lyrics were classified correctly. However, in the variable-artist analysis, 16.6% of country songs were classified as rock and 26.5% of rock songs were classified as country; with single artists for each genre, 17.7% of country songs were classified as rock (similar to previous result) but only 17.9% of rock songs were classified as country (much lower than 26.5%). This increase in rock genre classification accuracy seems to account for a majority of the overall accuracy improvement.

The same-genre artist classifier performed surprisingly well, achieving 72.1% accuracy with SVM. Because both SVM and Maxent internally compute feature importance, it is very difficult to assess “what went wrong” on misclassified songs. From the SVM confusion matrix we can deduce that U2 songs differed the most from Blink-182 songs (only 6.8% of U2 songs were classified as Blink-182) and that Blink-182 songs were the hardest to classify (41.9% of Blink-182 songs were misclassified). It would be interesting to develop additional classification features that distinguish better between different artists (since our initial feature set aimed to distinguish between genres), but we didn’t have time to explore this.

When lyrics of all five artists were classified together, Maxent achieved 70.1% classification accuracy. Considering the baseline accuracy for a five-group classifier is much lower than that of a three-group classifier (20% < 33%), this was a rather pleasing result. Once again, rap songs (Snoop Dogg) were easiest to classify, while Blink-182 songs were the hardest. Interestingly, Garth Brooks’s songs were more often misclassified as Beatles or U2 songs than as Blink-182 songs (perhaps due to other factors such as target audience, release date, etc.).

V. Conclusion and Future Work

Commentary

Although we only had several weeks to work on this project, we feel we produced some interesting and promising results. Using a limited feature set derived from song lyrics and absolutely no acoustic information, we were able to classify over 81% of song lyrics correctly. With more time, we would have liked to enhance our classifier with the following:

- Additional training data. It was difficult to obtain a sufficiently large corpus of training data using the EvilLyrics utility. If we had more training data available, certain features such as PoS would have more consistent results. Furthermore, we could try more refined features. For example, instead of looking at average line length in SVM, we could look at a histogram-style distribution over several buckets of different lengths, and compute statistics over each individually.
- Our features incorporated no knowledge of word meaning. It would have been interesting to use a database such as WordNet to determine the general (broad) categories that words in a song fall into (for example, rock songs tend to talk about love and relationships, while country songs tend to be more story oriented in their focus, often recalling past events).
- Our punctuation feature simply counts the total number of punctuation marks that appear in a song; it would be interesting to refine this feature by making separate counts for different types of punctuation marks (for example, country songs tend to have many apostrophes).
- As previously mentioned, it would also be interesting to split up the style and content sections of word bags. Function words tend to represent author style, and as such could be used to indirectly distinguish genres through similar artist tendencies. Content words indicate song topic, and can be used to split genres based on their themes. By splitting these, we will have more specific word bags which may allow for better classification.

VI. Sources

[1] Li, T. and Ogihara, M. Semi-supervised learning for music artists style identification. Proceedings of the Thirteenth ACM conference on Information and knowledge management. November 08-13, 2004. Washington, D.C., USA

[2] Luyckx, K. and Daelemans, W. Shallow Text Analysis and Machine Learning for Authorship At-tribution. CNTS-Language Technology Group, Univ. Antwerp, Belgium

[3] Dilts, R. "The Article of the Month: Transderivational Morphology." Accessed 6/1/2006. 1999 Santa Cruz, CA. Online .

[4] Dhanaraj, R and Logan,B. Automatic Prediction of Hit Songs. 6th International Conference on Music Information Retrieval. September 11-15, 2005. ISMIR 2005 (pp. 488-491). London, UK

[5] BBC http://www.bbc.co.uk/radio1/chart/

[6] Billboard http://www.billboard.com/bbcom/index.jsp

[7] Amazon http://www.amazon.com/

[8] EvilLyrics http://www.evillabs.sk/evillyrics/

[9] Weka http://www.cs.waikato.ac.nz/ml/weka/

[10] Stanford Log-linear Part-Of-Speech Tagger http://nlp.stanford.edu/software/tagger.shtml

[11] MontyLingua http://web.media.mit.edu/~hugo/montylingua/

Word/Importance	country/rap	rap/rock	country/rock
when	0.011556001	0.025997469	0.037553469
the	-0.05319686	0.012405242	-0.040791618
and	0.031332314	0.012406243	0.043738557
of	-0.025549934	-0.021938687	-0.047488621
is	-0.097079992	0.043116415	-0.053963577
i	0.032726422	0.038717051	0.071443473
a	0.046281704	-0.015911002	0.030370702
in	0.063784739	-0.039883472	0.023901266
that	0.036213203	0.026202601	0.062415804
me	0.045656258	-0.057111438	-0.01145518
more	0.043181143	0.002692053	0.045873196
away	0.013507389	-0.064631248	-0.051123859
they	-0.037601215	0.045882331	0.008281116
ive	0.038539554	-0.020663778	0.017875776
but	-0.0176659	0.051656643	0.033990744
just	0.038203369	0.011915997	0.050119367
im	-0.027425859	-0.021380919	-0.048806778
with	-0.048054362	0.019268259	-0.028786103
it	-0.049650658	0.048649021	-0.001001637
if	-0.035541182	-0.021239297	-0.056780479
cause	-0.005077274	0.042685735	0.037608461
do	0.025978082	0.02213619	0.048114272
get	-0.019528186	0.036912466	0.01738428
like	-0.043953987	0.06361884	0.019664853
baby	0.071022198	0.024244583	0.095266781
will	-0.007269547	-0.028033506	-0.035303053
her	0.04788102	-0.038256779	0.009624242
she	0.047149638	-0.023864021	0.023285617
so	-0.080546792	0.029584203	-0.050962589
or	-0.037950482	0.013633142	-0.02431734
thats	0.017312252	0.021282223	0.038594475
day	0.05373738	-0.01238409	0.04135329
take	0.027094648	0.027206306	0.054300954
oh	0.039668967	-0.042916107	-0.003247141
old	0.044086847	-0.008505193	0.035581655
days	0.027513542	0.01048911	0.038002652
your	-0.026591728	-0.021983855	-0.048575583
gonna	0.055635131	-0.034450445	0.021184685

Table 1. Bag of Words

Importance measure for each word and label pair. Note that for a pair of genres [genre1/genre2], a positive importance measure means the word “votes” for genre1 while a negative importance measure indicates the word “votes” for genre2.

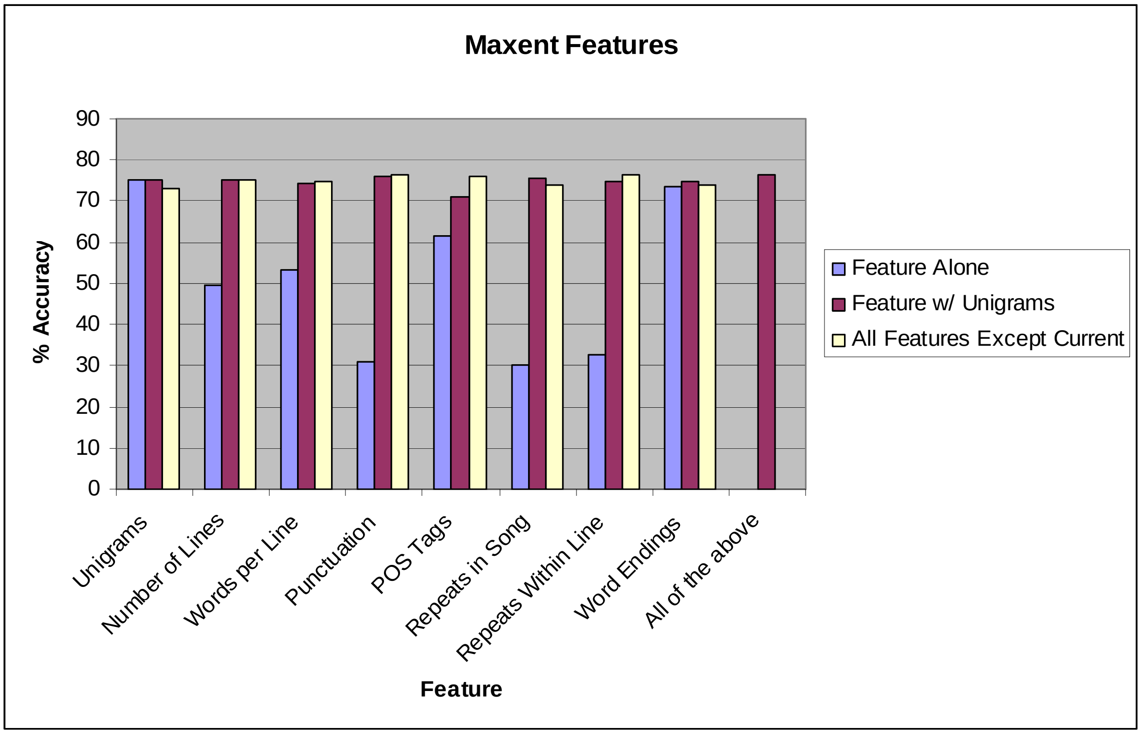
Ending/Importance	country/rap	rap/rock	country/rock
the	-0.061085968	0.011520055	-0.049565913
and	0.05542864	-0.008394705	0.047033935
ind	0.060915023	-0.02933998	0.031575044
ing	0.098790227	-0.11254773	-0.013757503
ine	0.036237059	-0.030105735	0.006131324
ome	0.035208	0.007792166	0.043000167
hat	0.026132833	0.028366539	0.054499371
ake	0.013822371	0.055405638	0.069228009
ore	0.037078624	-0.009168216	0.027910408
way	0.03522077	-0.060846724	-0.025625954
are	-0.016853199	-0.033561918	-0.050415117
ack	-0.019991736	0.046823929	0.026832194
hey	-0.040955417	0.042367621	0.001412204
ion	-0.038526828	0.010977296	-0.027549533
ive	0.024773577	-0.045124137	-0.02035056
but	-0.018502936	0.052273192	0.033770256
old	0.042681552	0.000451475	0.043133026
get	-0.013064521	0.04655824	0.03349372
ill	0.026843434	-0.057724814	-0.030881381
ugh	0.036508844	0.023508565	0.060017409
ith	-0.044056729	0.018669387	-0.025387342
own	0.06704978	-0.019222159	0.047827621
ave	0.024213533	-0.040818093	-0.016604559
ght	0.059872176	-0.01153078	0.048341396
use	-0.044076643	0.07739753	0.033320886
elf	-0.029918198	-0.021897332	-0.05181553
ver	0.045383406	0.043996056	0.089379463
ike	-0.047534473	0.066746391	0.019211918
aby	0.083539095	0.035227669	0.118766765
she	0.047149638	-0.023864021	0.023285617
ide	0.009210894	-0.042959655	-0.033748761
ats	-0.005794058	0.039725465	0.033931407
ere	0.058730673	-0.022449097	0.036281576
day	0.082873917	-0.035624757	0.04724916
our	-0.01607757	-0.024808334	-0.040885903
ong	0.036181521	-0.032398755	0.003782766
ain	-0.011687156	-0.030932661	-0.042619817
hes	0.042827925	-0.021294079	0.021533846
ven	0.014671999	0.022335623	0.037007622
ted	-0.032721729	-0.022862966	-0.055584695
nna	0.066247338	-0.019844647	0.046402691
art	0.038389668	0.019850009	0.058239677
ove	0.047302316	0.019329778	0.066632094
els	0.027536486	0.01210882	0.039645305
eah	0.048043772	-0.031275149	0.016768622
tle	0.040556236	-0.012930759	0.027625477

Table 2. Bag of Word Endings: Importance measure for each ending and label pair

Maxent			
Features	**Feature Alone**	**Feature w/ Unigrams**	**All Features Except Current**
Bag of Words -Unigrams	74.98	74.98	73.03
Number of Lines	49.7	74.97	75.31
Words per Line	53.33	74.39	74.74
Punctuation	30.93	76.08	76.26
POS Tags	61.46	71.15	76.07
Repeats in Song	30.17	75.69	73.99
Repeats Within Line	32.43	74.78	76.26
Word Endings	73.43	74.59	73.79
All of the above	NA	76.45	NA
SVM			
SVM Feature	**Feature Alone**	**Feature w/ Bag of Words**	**All Features Except Current**
Bag of Words	72.49	72.49	77.04
Number of Lines	54.65	73.25	80.27
Words per Line	36.62	73.055	81.02
Punctuation	37	73.25	80.83
POS Tags	61.29	75.71	79.51
Repeats in Song	43.64	76.28	81.02
Repeats Within Line	37.76	73.43	80.65
Word Endings	72.68	76.85	77.99
All of the above	NA	81.21	NA

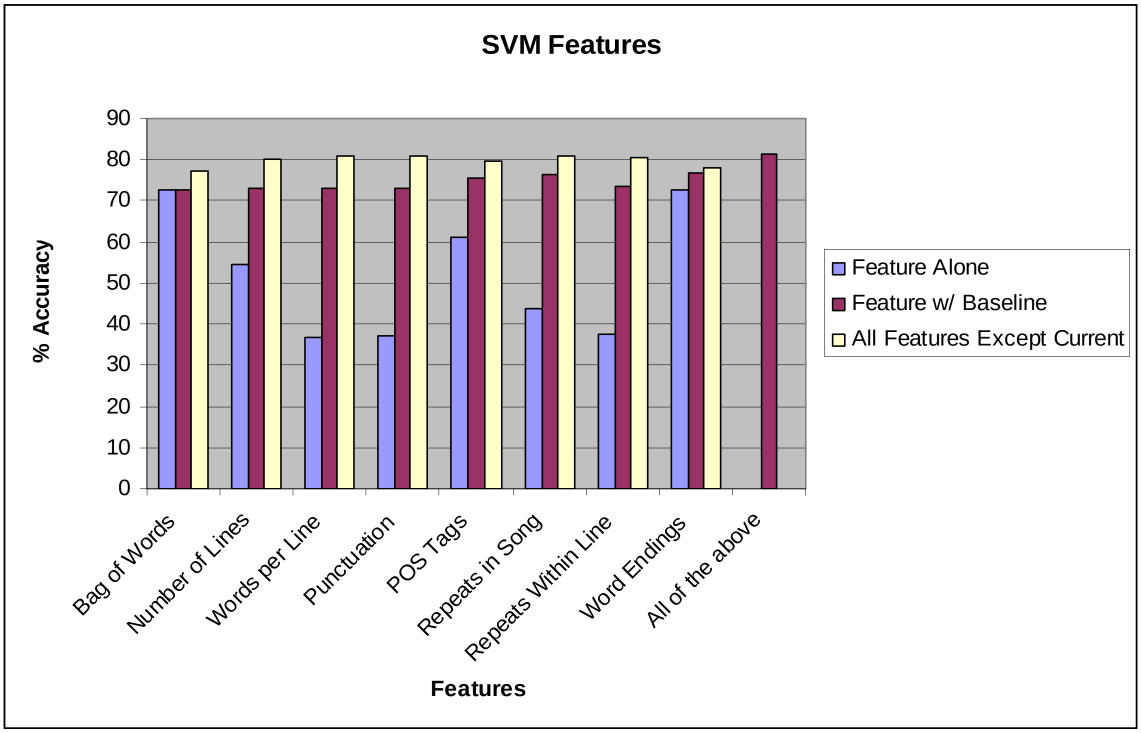
Table 3. Feature

% Accuracy on classifying genres {country, rap, rock} using 10-fold cross-validation for different combinations of features using Maxent and SVM.



Graph 1. Maxent Features

% Accuracy on classifying genres {country, rap, rock} using 10-fold cross-validation for different combinations of features using Maxent.



Graph 2. SVM Features

% Accuracy on classifying genres {country, rap, rock} using 10-fold cross-validation for different combinations of features using SVM.