Rhyme, Rhythm, and Rhubarb

Using Probabilistic Methods to Analyze Hip Hop, Poetry, and Misheard Lyrics

by

Hussein Hirjee

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Mathematics
in
Computer Science

Waterloo, Ontario, Canada, 2010

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

While text Information Retrieval applications often focus on extracting semantic features to identify the topic of a document, and Music Information Research tends to deal with melodic, timbral or meta-tagged data of songs, useful information can be gained from surface-level features of musical texts as well. This is especially true for texts such as song lyrics and poetry, in which the sound and structure of the words is important. These types of lyrical verse usually contain regular and repetitive patterns, like the rhymes in rap lyrics or the meter in metrical poetry. The existence of such patterns is not always categorical, as there may be a degree to which they appear or apply in any sample of text. For example, rhymes in hip hop are often imperfect and vary in the degree to which their constituent parts differ. Although a definitive decision as to the existence of any such feature cannot always be made, large corpora of known examples can be used to train probabilistic models enumerating the likelihood of their appearance. In this thesis, we apply likelihood-based methods to identify and characterize patterns in lyrical verse. We use a probabilistic model of mishearing in music to resolve misheard lyric search queries. We then apply a probabilistic model of rhyme to detect imperfect and internal rhymes in rap lyrics and quantitatively characterize rappers' styles in their use. Finally, we compute likelihoods of prosodic stress in words to perform automated scansion of poetry and compare poets' usage of and adherence to meter. In these applications, we find that likelihood-based methods outperform simpler, rule-based models at finding and quantifying lyrical features in text.

Acknowledgements

I would like to thank first and foremost my advisor Professor Daniel G. Brown for his guidance, honesty, and openness to weird and wonderful research directions, for his generous patience with my work habits and my short memory for statistical principles, and for generally being a pretty cool guy. Quick to offer positive feedback and suggest interesting areas of study, I would always leave his office feeling much more enthusiastic about my research than when I came in. I cannot imagine having created anything like this work without his influence and encouragement.

I am appreciative of the attention and assistance I have received from all of the staff and faculty at the university, and I want to especially thank my readers Professors Michael Terry and Charlie Clarke for graciously offering their time and energy to help me finish my degree.

I am, of course, extremely grateful for David Cheriton's contribution to the School of Computer Science and the Canadian government's NSERC program, whose financial support has allowed me to get paid for doing something I actually enjoy talking about.

Finally, I would like to thank my family for their constant love and support; especially my parents, for being OK with me studying rhymes; and my friends for distracting me from listening to music and reading poetry all day; in particular Liban, for suggesting interesting rhyme features; Daniel, for helping with the misheard lyric search indexing; Tanya, for giving me the Run-D.M.C. book; and David, for genuinely appreciating rap lyrics.

Dedication

"Yeah, this thesis is dedicated to all the teachers that told me I'd never amount to nothin', to all the people that lived above the buildings that I was hustlin' in front of that called the police on me when I was just tryin' to make some money to feed my daughter, and all the studentz in the struggle, you know what I'm sayin?" [13]

Contents

List of Tables

List of Figures

Nomenclature

affricate A complex phoneme consisting of a plosive followed by a fricative.

anapest A foot with two lightly stressed syllables followed by a heavy one.

approximant A phoneme produced by narrowing, without obstructing, the vocal tract.

dactyl A foot with one heavily stressed syllabble followed by two light ones.

diphthong A complex phoneme which begins as one vowel but ends as another.

foot The basic repeating stress unit in a metrical poem.

fricative A phoneme produced by breath being forced through a partially obstructed passage.

iamb A foot with one lightly stressed syllable followed by a heavy one.

IPA The International Phonetic Alphabet.

meter The regular, underlying stress pattern in a poem.

phoneme The fundamental unit of speech sound in a language.

plosive A phoneme produced by the stoppage and sudden release of breath.

prosody The pattern of intonation and stress applied to syllables in language.

scansion The assignment of meter to poetic verse.

trochee A foot with one heavily stressed syllable followed by a light one.

Chapter 1

Introduction

In Music Information Research, investigators tend to focus on extracting melodic or timbral data from audio signals to characterize and categorize the musical sound of songs, or on aggregating user-generated meta-tags and ratings to recommend similar music and generate playlists. In the text Information Retrieval domain, applications often involve extracting semantic features using the meanings and grammatical classification of words to generate information about the topical content and writing style of a document. When song lyrics are studied, it is generally their content which is mined for mood and theme indicators. This work is our attempt to synthesize these fields of study with methodology derived from biological sequence analysis to investigate a neglected aspect of text: the sound of the words themselves. These speech sounds are especially important in texts of a musical or poetic nature, *i.e.* lyrical verse.

When music listeners attempt to identify and retrieve songs, they often perform searches using the lyrics that they have heard. However, the lyrics that they hear are not always the lyrics that were sung, preventing standard text search engines from retrieving the desired song. The sound of the lyrics is far more important than their meaning for this task.

In hip hop music, rappers perform intricately rhymed verses over repetitive instrumental beats. Due to the relative simplicity of the backing tracks, most of the musical content in rap songs is contained within the lyrics themselves. Rappers' inventive formation, usage, and positioning of rhymes within lines of lyrics form an integral part of their unique style, about which they often boast.

In conventional poetry, poets often use the prosodic stress patterns within words to produce works with rhythmic meter underlying the structures. The less conventional among them modify these meters to create more complex metrical forms. Furthermore, they deliberately use words that do not fit the underlying meter to emphasize emotionally charged and figurative language.

In this thesis, we apply probabilistic techniques from biological sequence analysis to lyrical texts to identify, define, and discover these patterns and features.

1.1 The Theory

The main inspiration for this work comes from the field of bioinformatics. In the past few decades, vast amounts of biological data have been compiled as the DNA and RNA of thousands of organisms is continually being sequenced. A common application in the field is the comparison of biological sequences using alignments, in which unknown information about one sequence can be inferred from data known about the other. This allows for the prediction of gene expression and function, protein structure, and possible targets for drug discovery [20, 104].

While the goal in many sequence comparison tasks is to determine the probability that a given pair of sequences is related, it is often impossible to calculate this probability directly. The common solution to this problem is to use Bayesian methods to estimate these likelihoods. The basic methodology requires a collection of pairs of sequences known to be related. This collection is used to train a probabilistic model of similarity which provides a log-odds score for any pair of characters indicating their likelihood of being matched in related sequences vs. randomly. This score compares the frequency with which the pair of characters is seen matched in alignments of similar sequences with the background frequency of the pair in the collection.

Suppose we have an alignment of two protein sequences A and B, and we would like to know whether they are homologous (meaning that they are from species evolved from a common ancestor). What we want is the posterior probability of the hypothesis that they are homologous (H), $\Pr(\mathrm{H}|A, B)$, but what we can more easily calculate is the likelihood of seeing the particular alignment of A and B given a homologous pair of sequences: $\Pr(A, B|\mathrm{H})$. Using Bayes' Theorem, the probability of the hypothesis becomes:

$$\Pr(\mathrm{H}|A, B) = \frac{\Pr(A, B|\mathrm{H})}{\Pr(A, B)}[30] \tag{1.1}$$

Since the probability of the pair being homologous alone is not particularly informative, it must be compared with the probability that the alignment is due to the proteins matching by chance (R). An odds ratio is used to make the comparison:

$$\frac{\Pr(\mathrm{H}|A, B)}{\Pr(\mathrm{R}|A, B)} = \frac{\Pr(A, B|\mathrm{H})/\Pr(A, B)}{\Pr(A, B|\mathrm{R})/\Pr(A, B)} = \frac{\Pr(A, B|\mathrm{H})}{\Pr(A, B|\mathrm{R})} \tag{1.2}$$

Under the assumption that the component amino acids (i.e. the "letters") of the sequences are independent (which is not necessarily true, but useful in this model), the probability of seeing the alignment is the product of the probabilities of each constituent amino acid pair:

$$\Pr(A, B|\mathrm{H}) = \prod_i \Pr(A_i, B_i|\mathrm{H}), \text{ and} \tag{1.3}$$

$$\Pr(A, B|\mathrm{R}) = \prod_i \Pr(A_i, B_i|\mathrm{R}). \tag{1.4}$$

$\Pr(A_i, B_i|\mathrm{H})$, the likelihood of seeing amino acids A_i and B_i matched in a homologous sequence, is taken to be the frequency of A_i,B_i pairs in collections of proteins known to be homologous. $\Pr(A_i, B_i|\mathrm{R})$, the likelihood of seeing the particular amino acids match by chance, is taken to be the product of the background frequencies of A_i and B_i in the data.

When the logarithm of the odds ratio is taken, the constituent parts can be added together:

$$\log_2 \frac{\Pr(\mathrm{H}|A, B)}{\Pr(\mathrm{R}|A, B)} = \sum_i \log_2 \frac{\Pr(A_i, B_i|\mathrm{H})}{\Pr(A_i, B_i|\mathrm{R})} = \sum_i \mathrm{s}(A_i, B_i), \tag{1.5}$$

resulting in a log-odds score $\mathrm{s}(A_i, B_i)$ for each pair of amino acids, and a summed total score for the alignment. A positive score indicates that the alignment is more likely to be for homologous proteins; a negative score indicates that the alignment is seen by chance; a score of zero means that both hypotheses are equally likely. The individual amino acid scores can be grouped into a scoring matrix determining the likelihood of any pair matching in homologous vs. randomly matched proteins.

1.2 The Applications

With the advent and growth of the internet, the compilation of large and accessible corpora of texts has become easily achievable and various websites now exist devoted to various kinds of song lyrics, poetry, musical scripts, and other literature. This suggests the application of probabilistic methods derived from bioinformatics for the purposes of lyrical sequence analysis: automated processing of large collections of text can be used to train probabilistic models describing many different features of lyrical text. Based on the data used to train them, these models can be tailored specifically for particular features and can then be used to identify them in other documents and make inferences and characterizations about their existence in different corpora. The general approach in this work involves producing alignments of lyrical texts to generate log-odds scoring matrices of different constituent parts (letters, phonemes, syllables) and for different features (acoustic similarity, rhyme, and metrical stress).

In Chapter 2, we introduce a probabilistic model of mishearing sung lyrics. This model is trained using alignments of phonetic transcriptions of actual misheard lyrics with their correct counterparts. We then use this model's log-odds likelihood scores to perform phoneme alignment pattern matching to search for the correct lyric from a collection given a misheard lyric query. We compare the performance of this model with simpler, rule-based methods on a set of 146 misheard lyric queries with correct target lyrics in a collection of 2,345 songs. We then identify and describe queries for which the correct target lyrics were not found in the collection. Finally, we build and evaluate a phoneme trigram indexing system to speed up query run times by avoiding exhaustive dynamic programming search.

In Chapter 3, we develop a model of rhyme in hip hop music. This model is trained using syllable-by-syllable alignments of end rhymes from a corpus of influential rap lyrics in rhyming couplets. We use this model's syllable pair similarity scores to detect both internal and imperfect rhymes in a collection of lyrics and calculate statistical features about these detected rhymes. We then analyze these features and investigate their relationships with time, critical acclaim, and commercial success. We use rhyme feature-based classification to characterize and compare style in hip hop, and suggest other applications of this stylometry. Finally, we present a user interface which allows for the simple visualization and analysis of rhyme in rap lyrics.

In Chapter 4, we present an algorithm to perform automatic scansion and produce a correspondence score to quantify how well a poem aligns with the underlying metrical structure. We calculate likelihoods of metrical stress in words using alignments of prosodic stress markings with template metrical stress patterns discovered in a large collection of poems. We then use these likelihoods to develop a probabilistic method of performing scansion that allows for less strict metrical forms. Next, we compare different poets' adherence to metrical form and examine the emotional content and imagery of words on beats with modified stress. We then compare the formation of rhyme between hip hop and conventional poetry. Finally, we present a user interface tool which performs phonetic transcription, and scansion analysis and visualization.

Chapter 2

Solving Misheard Lyric Queries

2.1 Introduction

In this chapter, we apply probabilistic pattern-matching techniques inspired by algorithms from bioinformatics to resolve misheard song lyric queries. Though most Music Information Research (MIR) work on music query and song identification is driven by audio similarity methods, in which features extracted from digital signals are compared [67, 87, 71], users often use lyrics to determine the artist and title of a particular song, such as one they have heard on the radio. A common problem occurs when the listener either mishears or misremembers the lyrics of the song, resulting in a query that *sounds* similar to, but is not the same as, the actual words in the song she wants to find.

Furthermore, entering such a misheard lyric query into a search engine often results in many practically identical hits caused by various lyric sites having the exact same versions of songs. For example, a Google search for “Don’t walk on guns, burn your friends” (a mishearing of the opening line, “Load up on guns and bring your friends,” from Nirvana’s “Smells Like Teen Spirit”) gets numerous hits to “Shotgun Blues” by Guns N’ Roses (Figure 2.1). A more useful search result would give a ranked list of possible matches to the input query, based on some measure of similarity between the query and text in the songs returned. This goal suggests a similarity scoring measure for speech sounds: which potential target lyrics provide the best matches to a misheard lyric query?

The misheard lyric phenomenon has been recognized for quite some time. Sylvia Wright coined the autological term “Mondegreen” in a 1954 essay. This name refers to the lyric “They hae slain the Earl O’ Moray / And laid him on the green,” misheard to include the murder of one “Lady Mondegreen” as well [112]. However, the problem has only recently been tackled in the MIR community.

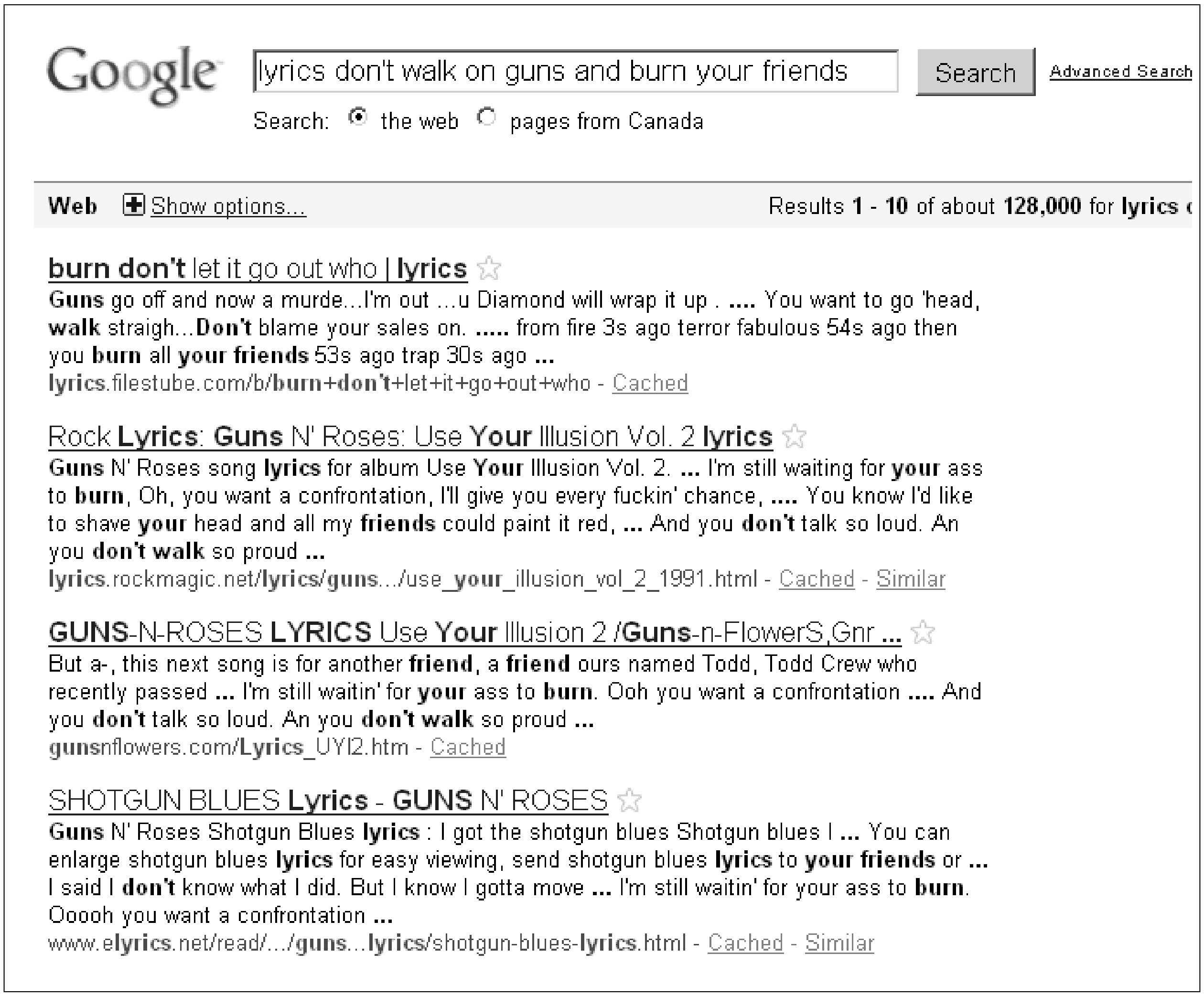

Figure 2.1: Search for misheard lyrics from "Smells Like Teen Spirit" returning results for Guns N' Roses.

Ring and Uitdenbogerd [94] compared different pattern-matching techniques to find the correct target lyric in a collection given a misheard lyric query. They found that a method based on aligning syllable onsets performed the best at this task, but the increase in performance over simpler methods was not statistically significant. Xu et al. [113] developed an acoustic distance metric based on phoneme confusion errors made by a speech recognition program. Using this scoring scheme provided a slight improvement over phoneme edit distance; both phonetic methods significantly outperformed a standard text search engine.

We describe a probabilistic model of mishearing based on phonetic confusion data derived from pairs of *actual* misheard and correct lyrics found on misheard lyrics websites. For any pair of phonemes a and b, this model produces a log-odds score giving the likelihood of a being (mis)heard as b. We replicated Ring and Uitdenbogerd's experiments using this model, as well as phonetic edit distance as described in Xu et al.'s work, on misheard lyric queries from the misheard lyrics site KissThisGuy.com. Our statistical method significantly outperformed all other techniques, and found up to 8% more correct lyrics than phonetic edit distance. Our work is presented in ISMIR 2010 [49].

2.2 Related Work

Ring and Uitdenbogerd [94] compared three different pattern-matching techniques for finding the correct lyrics or matches judged to be relevant given a misheard lyric query. The first is a simple Levenshtein edit distance [65] performed on the unmodified text of the lyrics. The second, Editex, groups classes of similar-sounding letters together and does not penalize substitutions of characters within the same class as much as ones not in the same class.

The third algorithm is a modified version of Syllable Alignment Pattern Searching they call SAPS-L [41]. In this method, the text is first transcribed phonetically using a set of simple text-to-phoneme rules based on the surrounding characters of any letter. It is then parsed into syllables, with priority given to consonants starting syllables (onsets). Pattern matching is performed by local alignment where matching syllable onset characters receive a score of +6, mismatching onsets score -2, and other characters score +1 for matches and -1 for mismatches. Onsets paired with non-onset characters score -4, encouraging the algorithm to produce alignments in which syllables are matched before individual phonemes. SAPS is especially promising since it is consistent with psychological models of word recognition in which segmentation attempts are made at the onsets of strong syllables [73].

They found that the phonetic based methods, Editex and SAPS-L, did not outperform the simple edit distance for finding all lyrics judged by assessors to sound similar to a given query misheard lyric but SAPS-L most accurately determined its single correct match.

However, due to the size of the test set of misheard lyric queries, they did not establish statistical significance for these results.

In a similar work, Xu et al. [113] first performed an analysis of over 1000 lyric queries from Japanese question and answer websites and determined that 19% of these queries contained misheard lyrics. They then developed an acoustic distance based on phoneme confusion to model the similarity of misheard lyrics to their correct versions. This metric was built by training a speech recognition engine on phonetically balanced Japanese telephone conversations and counting the number of phonemes confused for others by the speech recognizer. They then evaluated different search methods to determine the correct lyric in a corpus of Japanese and English songs given the query misheard lyrics. Phonetic pattern matching methods significantly outperformed Lucene, a standard text search engine. However, their acoustic distance metric only found 2-4% more correct lyrics than a simpler phoneme edit distance, perhaps due to its basis on machine speech recognition. They also implemented an indexed version of the search which reduced the running time by over 85% with less than 5% loss of accuracy.

2.3 Method

2.3.1 A Scoring Approach

We used a model inspired by protein homology detection techniques from bioinformatics, in which proteins are identified as sequences of amino acids. In this framework, a pair of proteins is modeled as two sequences of amino acid symbols generated either randomly or based on shared ancestry (known as homology) [30]. Using the BLOSUM (BLOcks of amino acid SUbstitution Matrix) local alignment scoring scheme, pairs of amino acids are assigned log-odds scores based on the likelihood of their being matched in alignments of homologous proteins. A positive score indicates the pair more likely co-occurs in proteins evolved from a shared ancestor, while a negative score indicates the pair is more likely to co-occur due to chance [46]. In a BLOSUM matrix M, the score for any two amino acids i and j, is calculated as

$$M[i,j] = \log_2 \frac{\Pr(i,j|\mathrm{H})}{\Pr(i,j|\mathrm{R})}, \tag{2.1}$$

where $\Pr(i,j|\mathrm{H})$ is the likelihood of i being matched to j in an alignment of two homologous proteins, while $\Pr(i,j|\mathrm{R})$ is the likelihood of them being matched by chance.

These likelihoods are calculated using frequencies of amino acid pairings in alignments of proteins known to be homologous. Given an amino acid residue pair frequency table A, where $A_{i,j}$ is the number of times residue i is matched to residue j in a collection of homologous protein alignments, the homology likelihood is calculated as

$$\Pr(i,j|\mathrm{H}) = \frac{A_{i,j}}{\sum_m \sum_n A_{m,n}}. \tag{2.2}$$

This corresponds to the proportion of amino acid pairs in which i matches with j. The match by chance likelihood is calculated as

$$\Pr(i,j|\mathrm{R}) = \frac{A_i \times A_j}{\sum_m A_m \times \sum_n A_n}, \tag{2.3}$$

where A_i is the total number of times amino acid i appears in the collection. This is simply the product of the background frequencies of each amino acid in the pair. If a pair of protein sequences contains regions in which the amino acids align to give high scores, the pair is considered to be homologous.

A similar methodology is employed by Ristad and Yianilos [95] for learning a stochastic string edit distance from a collection of examples. They found that this type of edit distance achieved close to one fifth of the error rate of a simpler Levenshtein edit distance at the task of determining the pronunciations of unlabeled words.

In the song lyric domain, we treat lines and phrases as sequences of phonemes and develop a model of mishearing to determine the probability of one phoneme sequence being misheard as another. This requires a pairwise scoring matrix which produces log-odds scores for the likelihood of pairs of phonemes being confused. The score for a pair of phonemes i and j is calculated as in Equation (2.1), where $\Pr(i,j|\mathrm{H})$ is the likelihood of i being heard as j, and $\Pr(i,j|\mathrm{R})$ is the likelihood of i and j being matched by chance. Instead of base 2, we use natural logarithms for these scores:

$$M[i,j] = \ln \frac{\Pr(i,j|\mathrm{H})}{\Pr(i,j|\mathrm{R})}, \tag{2.4}$$

In analogy to the likelihoods of pairs of amino acids that give rise to the BLOSUM matrix, these phoneme pair likelihoods are calculated using frequencies of phoneme confusion in actual misheard lyrics. Instead of an amino acid pair frequency table, we use a phoneme confusion frequency table F, where $F_{i,j}$ is the number of times a phoneme i is heard as j (where j may equal i). As in Equation 2.2 for amino acids matching due to homology, the mishearing likelihood is calculated as

$$\Pr(i,j|\mathrm{H}) = \frac{F_{i,j}}{\sum\limits_m \sum\limits_n F_{m,n}}, \tag{2.5}$$

which is the proportion of phoneme pairs in which i is heard as j. Similarly, if F_i is the total number of times phoneme i appears in the lyrics, the match by chance likelihood is calculated as

$$\Pr(i, j|\mathrm{R}) = \frac{F_i \times F_j}{(\sum_m F_m)^2}, \tag{2.6}$$

which is the product of the background frequencies of each phoneme in the pair.

2.3.2 Training Data for the Model

To produce the phoneme confusion frequency table F, we required a training set of misheard lyrics aligned to their correct versions. Our corpus contains query and target pairs from two user-submitted misheard lyrics websites, KissThisGuy.com and AmIRight.com. In both cases, the first phrase in the pair is the song lyric heard by the submitter and the second phrase is the true song lyric.

The KissThisGuy.com pairs were provided by HumorBox Entertainment, the parent company of KissThisGuy.com, and consist of 9,527 pairs randomly selected from the database and comprising 10% of the total number of misheard lyrics on the website. The pairs from AmIRight.com were selected from the pages for the top 10 artists (by number of misheard lyrics submitted) on the site and total 11,261 pairs, roughly corresponding to 10% of the misheard lyrics on the site. The artists included are The Beatles, Michael Jackson, Elton John, Nirvana, Red Hot Chili Peppers, Queen, Metallica, Madonna, and Green Day, as well as traditional songs.

2.3.3 Producing Transcriptions

We first used the Carnegie Mellon University pronouncing dictionary to obtain phonetic transcriptions of the lyrics. The CMU pronouncing dictionary has phonetic transcriptions for over 100,000 words and is tailored for North American English [64], the language used by the majority of artists in our data. To avoid the complications and computational complexity required to evaluate all possible transcriptions for heteronyms and other words with numerous pronunciations, we selected the first transcription for each word, corresponding to the most common pronunciation.

The transcriptions contain 39 phonemes, consisting of 24 consonants, including affricates such as /tʃ/ and /dʒ/, and 15 vowels, including diphthongs like /aʊ/ and /ɔɪ/ [9]. The vowels include metrical stress markings indicating whether they receive primary (1), secondary (2), or no stress (0). Thus, for each word in the dictionary, the transcription provides the speech sounds (phonemes), as well as the prosody (the pattern of emphasis placed on each syllable when pronounced.) To avoid overfitting due to the relatively small number of secondary stressed syllables in the dictionary, we combined primary and secondary stresses into strong stress to contrast with weak or unstressed syllables. This resulted in a set of 54 phonemes: 24 consonants and 30 stress-marked vowels.

We used the Naval Research Laboratory's text-to-phoneme rules to transcribe any words not found in the dictionary [32]. These rules provide a phonetic substitution approximating the correct pronunciation for each of the 26 letters of the alphabet, based on the characters surrounding them in the word. The phonemes used in the NRL rules correspond almost exactly with those in the CMU dictionary; the only differences are a lack of stress markings and that /ə/ and /ʌ/ are treated as two separate phonemes, as opposed to the same phoneme with different assigned stress. The full list of phonemes used, along with their International Phonetic Alphabet (IPA) versions, and English examples is detailed in Table 2.1.

To better model actual prosody in singing, we reduced the stress in common single-syllable words with less metrical importance such as "a," "and," and "the." To allow for variation in the likelihood of different phonemes being missed (deleted) or misheard without having been sung (inserted), we introduced an additional symbol, "_", for gaps in alignment and treated it like any other phoneme. This allowed "softer" approximants such as /r/ to receive lesser penalties when missed than "harder" affricates such as /tʃ/.

2.3.4 Iterated Training

We performed an iterated alignment method with the lyric pairs to determine the confusion frequencies in the matrix F. In the first phase, phonemes were lined up sequentially starting from the left end of each phrase in the pair. This may seem to be too rough an alignment method, but it resulted in the highest frequencies for identical phoneme pairs since most of the misheard lyrics contained some correct lyrics within them. For example, "a girl with chestnut hair" being misheard as "a girl with just no hair" from Leonard Cohen's "Dress Rehearsal Rag" was first aligned as

A g—ir–l w–i–th j—u–s–t n–o h—ai—r
ə g 'ɝ l w ɪ θ dʒ 'ʌ s t n oʊ h 'eɪ r
ə g 'ɝ l w ɪ θ tʃ 'ɛ s t n ə t h 'eɪ r
A g—ir–l w–i–th ch—e–s–t—n–u—t h—ai—r

with all phonemes matching exactly until the /tʃ/ heard as /dʒ/, then the /'ɛ/ heard as /'ʌ/, etc. From these simple alignments, we constructed an initial phoneme confusion frequency table F'.

Since gaps do not appear explicitly in any lyrics, we approximated their occurrence by adding gap symbols to the shorter phrase in each pair to ensure that the phrases were of the same length. In the example above, we counted one gap, and had it occurring as an

CMU Phoneme	IPA Phoneme	Example Word
AA	ɑ	f**a**ther
AE	æ	**a**t
AH	ʌ / ə	h**u**t / **a**bout
AO	ɔ	**ough**t
AW	aʊ	c**ow**
AY	ai	h**i**de
B	b	**b**e
CH	tʃ	**ch**eese
D	d	**d**ee
DH	ð	**th**ee
EH	ɛ	**E**d
ER	ɝ / ɚ	h**ur**t / fath**er**
EY	ei	**a**te
F	f	**f**ee
G	g	**g**reen
HH	h	**h**e
IH	ɪ	**i**t
IY	i	**ea**t
JH	dʒ	**g**ee
K	k	**k**ey
L	l	**l**ee
M	m	**m**e
N	n	**kn**ee
NG	ŋ	pi**ng**
OW	oʊ	**oa**t
OY	ɔi	t**oy**
P	p	**p**ee
R	r	**r**ead
S	s	**s**ea
SH	ʃ	**sh**e
T	t	**t**ea
TH	θ	**th**eta
UH	ʊ	h**oo**d
UW	u	tw**o**
V	v	**v**ee
W	w	**w**e
Y	j	**y**ield
Z	z	**z**ee
ZH	ʒ	sei**z**ure [64]

Table 2.1: List of phonemes in CMU and IPA form

/r/ being missed in the F' table. This approximation resulted in an essentially random initial distribution of gap likelihood across phonemes.

Now, given the initial frequency table, we calculated an initial scoring matrix M' using Equations (2.1), (2.5), and (2.6) above. We then used the scores found in M' to align the pairs in the second phase of training. In this stage, we used dynamic programming [30] to produce the optimal global alignment between each misheard lyric and its corresponding correct version. This allowed for the inclusion of gaps in either sequence, regardless of its length. We then traced back through the alignment and updated the phoneme co-occurrences in a new confusion frequency table F. For the example cited above, the new alignment looked like

A g—ir–l w–i–th j—u–s–t n–o h—ai—r
ə g ˈɝ l w ɪ θ dʒ ˈʌ s t n oʊ _ h ˈeɪ r
ə g ˈɝ l w ɪ θ tʃ ˈɛ s t n ə t h ˈeɪ r
A g—ir–l w–i–th ch—e–s–t—n–u–t h—ai—r.

The gap was computed to have occurred earlier and resulted in an additional entry for $F[\_, /t/]$, increasing the frequency of missed /t/ phonemes. After all the pairs had been processed, we calculated a final scoring matrix M from frequency table F, as above.

2.3.5 Structure of the Phonetic Confusion Matrix

One interesting property of the phonetic confusion matrix is that, from first principles, we discovered perceptual similarities between sounds: if two phonemes a and b had positive scores in our confusion matrix, then they sounded similar to the real people who entered these queries into the misheard lyrics websites from which our database is drawn.

Table 2.2 shows all of the pairs of distinct consonant phonemes a and b such that $M[a, b]$ was positive. These consisted mainly of changes in voicing (*e.g.*, /g/ versus /k/) or moving from a fricative to a plosive (*e.g.*, /f/ versus /p/) The only distinct consonant pairs scoring above +1.0 (meaning they were at least $e^{1.0}$ or about 2.7 times more likely to appear in lyrics misheard as each other than expected by chance) were pairs of sibilants (such as /tʃ/ versus /dʒ/ or /ʒ/ versus /ʃ/). All of these similarities were discovered without any input knowledge or constraints on which phonemes should sound similar. They were discovered by the training process itself, and highlight the pairs which were confused for each other by actual music listeners.

Table 2.3 shows all pairs of distinct stressed vowel phonemes a and b such that $M[a, b]$ was positive. When examining these scores in detail, it becomes evident that vowel height is the least salient articulatory feature for listeners to determine from sung words, as most of the confused vowels differed mainly in height. These pairs included /ɑ/ and /ʌ/, /ʌ/ and /ʊ/, /æ/ and /ɛ/, and /ɛ/ and /ɪ/. Other common confusions included vowels differing

Query Phoneme	Target Phoneme
/b/	/f/,/p/,/v/
/tʃ/	/dʒ/,/k/,/ʃ/,/t/,/ʒ/
/f/	/b/,/p/,/θ/
/g/	/dʒ/,/k/
/dʒ/	/tʃ/,/ʃ/,/y/,/ʒ/
/k/	/g/
/ŋ/	/n/
/p/	/b/,/f/,/θ/,/v/
/s/	/z/
/ʃ/	/tʃ/,/dʒ/,/s/,/ʒ/
/θ/	/f/
/z/	/s/,/ʒ/
/ʒ/	/dʒ/,/ʃ/

Table 2.2: Non-identical consonants with positive scores.

mainly in length and diphthongs confused with their constituent phonemes, such as /ɪ/ with /i/, /ɑ/ with /aʊ/, and /ɔ/ with /ɔʊ/.

When examining differences in gap scores, we found that the phonemes most likely to be missed (deleted) or heard without being sung (inserted) were /r/, /d/, /ŋ/, and /z/. Although the model was trained without any domain knowledge, a semantic explanation is likely for this finding since /d/ and /z/ are often added to words to form past tenses or plurals which could be easily confused; for example, "jam" could easily be heard for both "jammed" or "jams." /ŋ/ is often changed to /n/ in verb present progressive tenses in popular music; for example, "runnin'" could be sung for "running." The phonemes least likely to be missed were /ʒ/, /ʃ/, /ɔɪ/, and /ɪ/, probably (with the surprising exception of /ɪ/) due to their relative "length" of sound. Similarly, /ʃ/, /ʊ/, /ɪ/, and /ɝ/ were least likely to be heard without being sung.

2.3.6 Searching Method

To perform phonetic lyric search with this model, we used matrix M to score semi-local alignments [30] between the query phrase (sequence of phonemes) and all candidate songs in the database. For every query phrase Q of length m and song T of length n, we constructed an $(m+1) \times (n+1)$ matrix S to score the alignment. After initializing the first row with zeros (to allow alignments to start anywhere in the song) and the first column with the penalties for inserting (hearing sounds that were not sung) the phonemes in Q, S was filled

Query Phoneme	Target Phoneme
/ɑ/	/ʌ/,/ɔ/,/aʊ/,aɪ/,oʊ/
/æ/	/ɑ/,/aʊ/,/aɪ/,/ɛ/
/ʌ/	/ɑ/,/ɔ/,/ɝ/,/oʊ/,/ʊ/
/ɔ/	/ɑ/,/ʌ/,/oʊ/,/ɔɪ/,/ʊ/
/aʊ/	/ɑ/,/æ/
/ɛ/	/æ/,/ɪ/
/ɝ/	/ʌ/,/ʊ/
/eɪ/	/ɛ/
/ɪ/	/ɛ/,/i/
/i/	/ɪ/
/oʊ/	/ɔ/,/ɔɪ/,/ʊ/
/ɔɪ/	/ɔ/,/oʊ/,/ʊ/
/ʊ/	/ɔ/,/ɝ/,/oʊ/,/ɔɪ/,/u/
/u/	/ʊ/

Table 2.3: Pairs of non-identical stressed vowels with positive scores.

in row-by-row and column-by-column as follows:

$$S[i,j] = \max \begin{cases} S[i-1,j-1] + M[Q[i-1],T[j-1]] & \text{(for matching/substituted phonemes)}, \\ S[i-1,j] + M[Q[i-1],\_] & \text{(for a phoneme being heard but not sung)}, \\ S[i,j-1] + M[\_,T[j-1]] & \text{(for a phoneme being sung but not heard)}. \end{cases} \tag{2.7}$$

After S was completely filled in, the highest value in the last row was selected as the best score for the alignment between the query phoneme string Q and the song text lyrics T. Selecting the maximum value from only the last row ensured that the entire query string was aligned. This differs from standard local alignment in which the maximum value in the entire scoring matrix is used as the best score.

In addition to this phonemic model, we developed a syllable-based model which produced a log-likelihood score for any syllable being (mis)heard as another. For any pair of syllables a and b, we calculated this score as

$$S[a,b] = \text{align}(a_o, b_o) + M[a_v, b_v] + \text{align}(a_e, b_e), \tag{2.8}$$

where a_v is the vowel in syllable a and $M[a_v, b_v]$ is defined in Equation 2.1 above. $\text{align}(a_o, b_o)$ is the score for the optimal global alignment between the onset consonants of a and b, and $\text{align}(a_e, b_e)$ is the score for the optimal global alignment between the ending consonants (or coda) of a and b.

Searching and training were performed in the same way as with the phonemic method (see Equation 4.1), except that syllables were aligned instead of phonemes. Essentially, this ensured that vowels only matched with other vowels and consonants only matched with other consonants.

2.4 Experiment

To compare the performance of the probabilistic model of mishearing with other pattern matching techniques, we reproduced the experiment of Ring and Uitdenbogerd [94] finding the best matches for a query set of misheard lyrics in a collection of full song lyrics containing the correct version of each query.

2.4.1 Target and Query Sets

We used Ring and Uitdenbogerd's collection, comprising a subset of songs from the lyrics site lyrics.astraweb.com containing music from a variety of genres by artists such as Alicia Keys, Big & Rich, The Dave Matthews Band, Queensrÿche, and XTC. After removing duplicates, it contained 2,345 songs with a total of over 486,000 words. This formed our set of targets.

We augmented their original query set of 50 misheard lyrics from AmIRight.com with 96 additional misheard lyrics from the KissThisGuy.com data. These additional queries had corresponding correct lyric phrases that matched exactly with a phrase from a single song in the collection. They did not necessarily match the same song the query lyric was misheard from, but only had one unique match in the collection. For example, "you have golden eyes" was heard for "you're as cold as ice," from Foreigner's "Cold As Ice," a song which does not appear in the collection. However, the same line occurs in 10cc's "Green Eyed Monster," which is in the collection. We included at most one query for each song in the collection. In practice, misheard lyric queries may have correct counterparts which appear in multiple songs, potentially making our results less generalizable for large corpora. However, this occurs for correct lyric queries as well, especially when they are short, common phrases like "I just want to" or "I love you," making it a general problem in lyric search, not one unique to the similarity-based matching for misheard lyrics.

2.4.2 Methods Used in Experiments

We implemented three different pattern-matching algorithms in addition to the probabilistic mishearing models described above: SAPS-L and simple edit distance as the best methods from Ring and Uitdenbogerd's paper, and phonemic edit distance to estimate a comparison with Xu et al.'s Acoustic Distance. (The actual scoring matrix used in that work was not made available to us despite our requests.) We removed all test queries from the training set for the probabilistic models.

2.4.3 Evaluation Metrics

For each method, we found the top 10 best matches for each misheard lyric in our query set and used these results to calculate the *mean reciprocal rank* (MRR_{10}) as well as the hit rate by rank for the different methods. The MRR_{10} is the average of the reciprocal ranks across all queries, where reciprocal rank is one divided by the rank of the correct lyric if it is in the top ten, and zero otherwise. The hit rate by rank is the cumulative percentage of correct lyrics found at each rank in the results.

As a small example illustrating these measures, consider a sequence of five queries where the correct lyrics are returned as the second entry, then the fourth entry, then the first entry, then not at all, then the ninth entry, respectively. The reciprocal ranks for this sequence would be $(\frac{1}{2}, \frac{1}{4}, 1, 0, \frac{1}{9})$, with an MRR_{10} of 0.372 (the average of this vector). The hit rate by rank would be 20% at 1 (one out of five correct lyrics returned as the first entry), 40% at 2 (two out of five returned in the first two entries), 40% at 3 (since no more correct lyrics were returned as the third entry), 60% at 4 through 8, and 80% at 9 and 10.

2.5 Results

The probabilistic model of phoneme mishearing significantly outperformed all other methods in the search task, achieving an MRR_{10} of 0.774 and ranking the correct lyric for 108 of the 146 queries (74.0%) first. The next best methods were phonemic edit distance and probabilistic syllable alignment, receiving MRR_{10} of 0.709 and 0.702, respectively. Performing a paired t-test on the reciprocal rankings of the probabilistic phoneme model and the phonemic edit distance returned a p-value less than 0.001, strongly indicating that the results were drawn from different distributions. There was no statistically significant difference between the probabilistic syllable model and the phonemic edit distance results. Both these methods were found to significantly outperform SAPS-L, with p-values less than 0.05 on paired t-tests. SAPS-L produced an MRR_{10} of 0.655, which was marginally better than the simple edit distance's MRR_{10} of 0.632. However, the difference between these two was again not found to be statistically significant. The Mean Reciprocal Rank results are shown in Table 2.4.

The hit rate by rank (Figure 2.2) further illustrates the effectiveness of the probabilistic phoneme model as it ranked between 5% and 8% more correct lyrics within the top five

Pattern Matching Method	Mean Reciprocal Rank
Probabilistic Phoneme Model	0.774
Phoneme Edit Distance	0.709
Probabilistic Syllable Model	0.702
SAPS-L	0.655
Simple Edit Distance	0.632

Table 2.4: Mean reciprocal rank after ten results for different search techniques.

matches than phonemic edit distance and the probabilistic syllable model. These next two methods appear to have performed equally well and considerably better than SAPS-L and edit distance. SAPS-L seems to improve in performance over simple edit distance moving down the ranks, indicating that it may be better able to find less similar matches.

2.5.1 Analysis of Errors

We also observed that the performance of the probabilistic phoneme model plateaued at a hit rate of 83%. This corresponds to 121 of the 146 misheard lyric queries, and we provide a brief analysis of some of the 25 queries missed.

Differences Among Methods

The phoneme edit distance method did not return any correct lyrics not found by the probabilistic phoneme model. There was one query for which SAPS-L returned a hit in the top 10 and the probabilistic model did not: "spoon aspirator" for "smooth operator," from Sade's song of the same name. In SAPS-L, this was transcribed as "SPoon AsPiRaTor," getting a score of 24 when matched with "Smooth OPeRaTor" (here, as in the SAPS-L transcription algorithm, syllable onsets are marked as upper case letters.) The relatively high number of matching syllable onsets (S, P, R, and T) in the short query gave SAPS-L the advantage since it heavily emphasizes matched onsets. On the other hand, the probabilistic method produced higher scores for results such as "spoon in spoon stir(ring)" and "I'm respirating" due to the high number of exactly matching and similar phonemes.

The probabilistic syllable model also returned a hit for one query for which the phoneme model did not. The misheard lyric in this case was "picture Mona runnin'" heard for "get your motor runnin'," presumably from Steppenwolf's "Born to be Wild." This was likely due to the parsing of the phonetic transcription so that paired syllables had high scores at both the onset and ending consonants ("Mon" and "mot", "run" and "run"). The top ranking match using the phoneme model was "picture on your button." When the

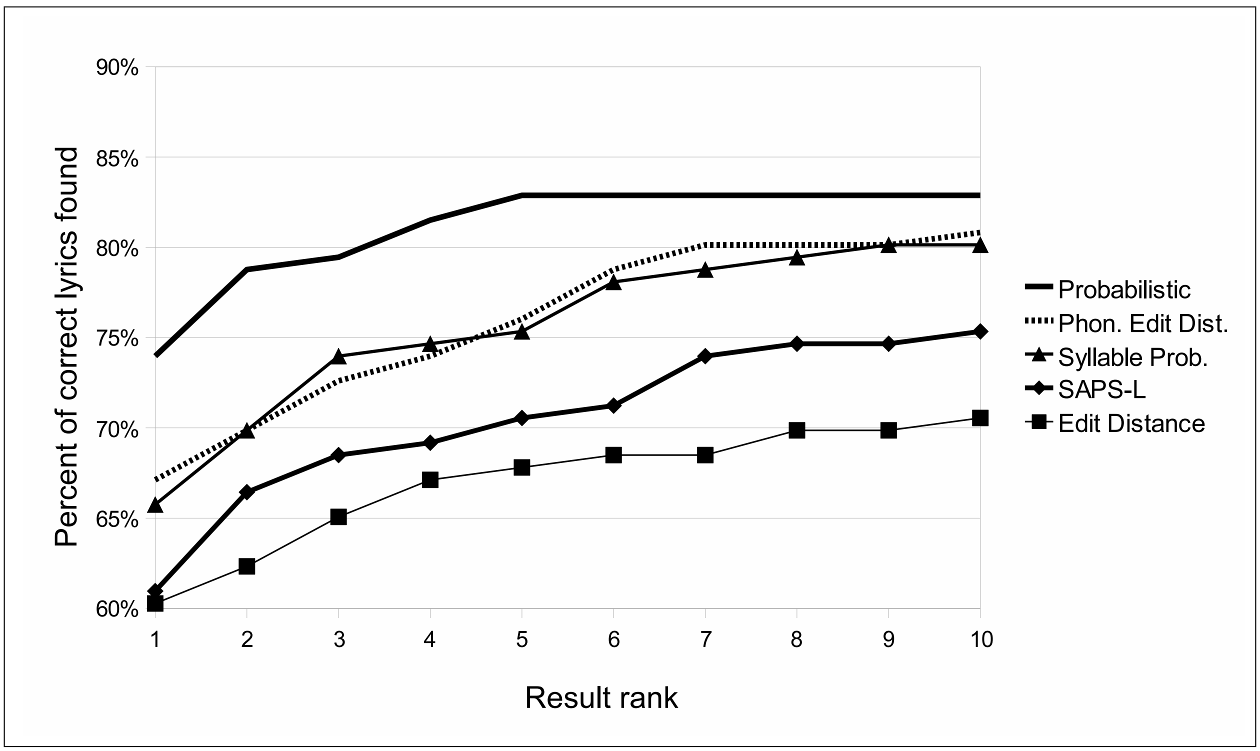

Figure 2.2: Cumulative percentage of correct lyrics found by rank for different search methods. The probabilistic phoneme model found 5-8% more correct targets in the first five matches than the next best method. The probabilistic syllable model and phoneme edit distance performed nearly identically, and significantly better than SAPS-L and simple edit distance.

phrases are transcribed without word or syllable boundaries, the only large differences are an inserted /m/ from “Mona” and a missed /b/ from “button.”

Common Types of Errors

Though syllable parsing and alignment may have helped for the two misheard lyrics described above, the majority of the queries not returning results tended to be quite dissimilar from their target correct lyrics. Some examples of these include a young child hearing “ooh, Tzadee, I’m in a cheerio” for “we are spirits in the material” from The Police’s “Spirits in the Material World;” “Girl, I wanna yodel” for “You’re The One That I Want” from Grease; “Apple, dapple, and do” for Prince’s “I Would Die 4 U;” “Swingin’ the bat” for the Bee Gees’ “Stayin’ Alive;” and the extremely puzzling “Rhubarb” for “Move out” from Yaz’s Situation. In other interesting cases the listener superfluously heard the singer’s name within the song lyrics: “Freddie time!” for “and turns the tides” in Queen’s My Fairy King (referring to lead singer Freddie Mercury), and “Oh, Lionel (Oh Line’)” for Lionel Richie’s “All Night Long (all night”). Without knowledge of the song’s performer, it would be difficult to consider these similar to their originals.

The other common problem preventing the algorithms from finding the correct matches for many misheard lyrics stems from the short length of such queries. Some of these included “chew the bug” for “jitterbug,” “can of tuna” for “can’t hurt you now,” “rhubarb” for “move out”, and “wow thing” for “wild thing.” While these tended to be fairly similar to their correct counterparts, their short length made it much more likely for exact partial matches to be found. These partially matching segments often scored highly enough to balance out the dissimilar remaining portions in incorrectly returned lyrics. Though the models were trained on mishearing, most misheard lyrics tend to have parts heard correctly, so matching identical phonemes will usually give the highest scores. For all the search methods, longer queries were more likely to have their correct lyrics found in the top 10, resulting in a positive correlation between the length of the query and the reciprocal rank of the correct result. Table 2.5 details these correlations for the different algorithms. This correlation is smallest for the probabilistic phoneme model, indicating that it is the least fragile in this way and best able to disambiguate even short misheard lyric queries.

2.6 Phonetic Indexing

The implementation of the search algorithm described so far is an exhaustive dynamic programming search over the entire collection of lyrics. This results in O(mn) computing complexity per query, where m is the length of the query and n is the size of the collection (in phonemes). This would likely not be feasible in a commercial application due to the

Pattern Matching Method	Correlation
Probabilistic Phoneme Model	0.45
Phoneme Edit Distance	0.54
Probabilistic Syllable Model	0.55
SAPS-L	0.53
Simple Edit Distance	0.51

Table 2.5: Correlation between misheard query length and reciprocal rank of correct answer returned. Longer queries were more likely to have the correct lyric ranked higher, though this effect was least pronounced for the probabilistic phoneme model.

long search time required (about 3 seconds per query on a 1.6 GHz laptop). Xu et al.[113] demonstrated the effectiveness of using n-gram indexing to reduce the running time by pre-computing the distances from 90% of all syllable trigrams in their collection and pruning off the most dissimilar lyrics. In a similar fashion, we experimented with trigram indexing on our collection. However, iterating syllable n-grams is simpler with Japanese than English since Japanese has a limited number of possible syllables (100). English, in contrast, has thousands of possible syllables. For this reason, we used phoneme n-grams in our indexing.

2.6.1 Building The Index

The method we followed in our indexing was similar to that of Xu et al. [113] and also inspired by the Basic Local Alignment Search Tool (BLAST) [6] used for protein and DNA homology searches within large databases of biological sequences. Both these methods involve creating an "inverted index," in which subsequences of the input query (called "words" or amino acid "k-mers" in the bioinformatics domain and n-grams in the MIR field), are used as the key to return a list of all highly similar "hits" in the database being searched. Using amino acid 4-mers in BLAST, for each of the 160,000 possible 4-mers (since there are 20 different amino acids for each of the four characters = 20^4) the entire database is first scanned to generate a list of all locations (hits) that score above a threshold T with that 4-mer. Then, for each 4-mer in a query sequence, the list of hits is returned and scanned. Alignments are extended in both directions from each hit until the similarity score drops a certain distance below the best score previously found for shorter alignments. This significantly reduces the overall running time since much less dynamic programming must be performed.

In building our index, we first iterated over all 175,616 (= 56^3) possible phoneme trigrams and for each we generated a list of every location in the lyric collection that scored above a threshold value T. To avoid having excessively long lists of hits which would drive

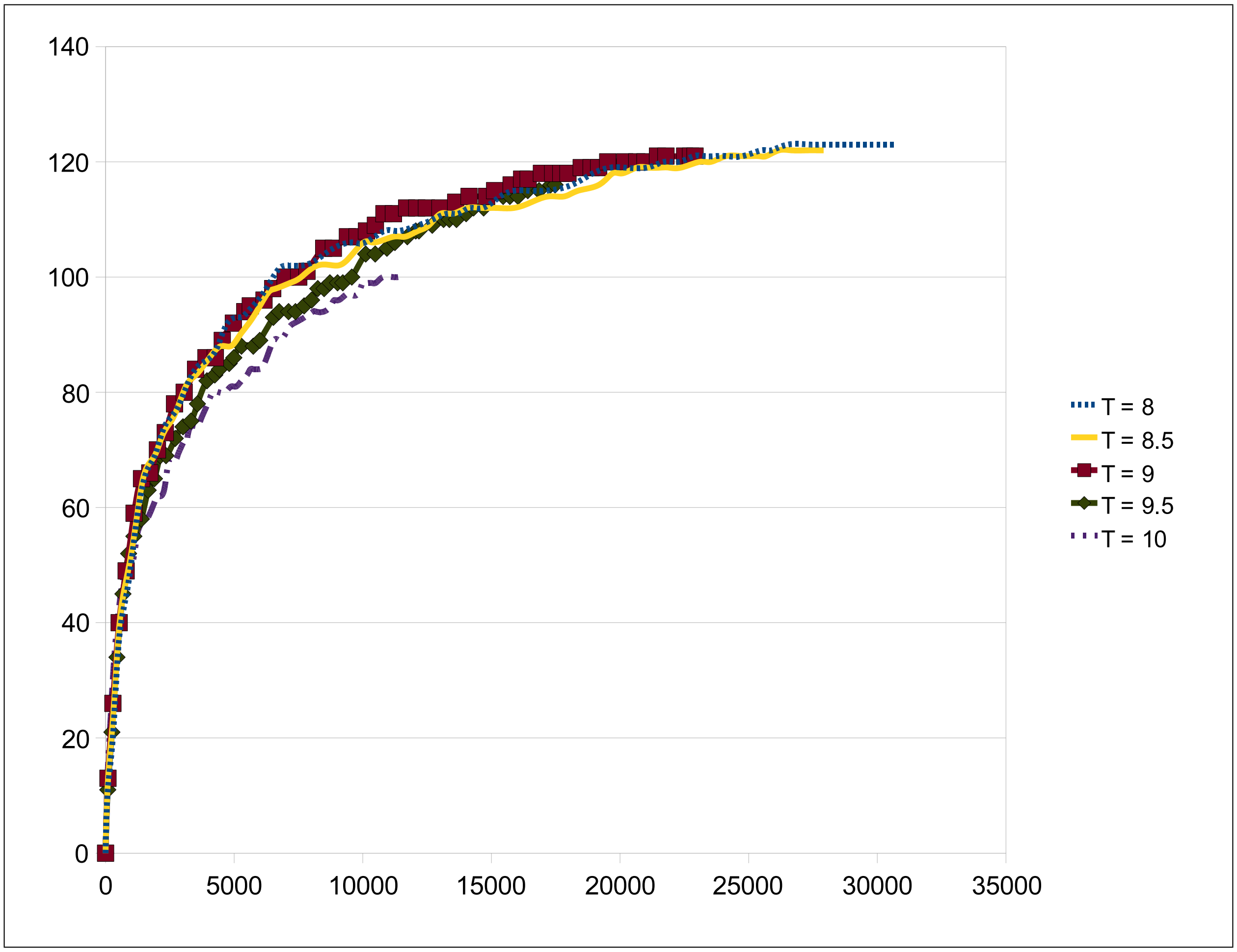

Figure 2.3: Number of correct lyrics hit versus total number of hits returned for different threshold values T. The threshold T = 9 performs as well as less strict T values.

up running time and memory requirements, we only saved hit lists for relatively uncommon trigrams. This naturally lead to the dilemma of determining what makes an uncommon trigram; *i.e.* how many hits should be allowed per trigram? We limited our lyric collection to only those songs for which we had a misheard query in our test set, and compared the total number of hits returned (as an estimation of required running time) with the number of queries returning a hit in the correct song lyric, while varying the maximum size hit list per trigram. We tried a few different values for the score threshold T and decided that 9 was a good value balancing sensitivity with specificity as higher values returned fewer correct results, but lower values didn't return many more. The comparisons between number of correct lyrics hit and total number of hits returned is illustrated in Figure 2.3.

All threshold choices had the number of correct lyrics hit plateauing in the high 120s (out of the 146 total queries), which is reasonable since only 121 songs were returned in the top ten when performing an exhaustive search. However, these results only indicate that the correct lyric had the *potential* to be returned in the top 10 if a search were performed with its query. Furthermore, we could not be certain that the query trigram causing the hit in the correct lyric necessarily matched the particular part of the song which had been (mis)heard as that trigram. Nevertheless, we used this score threshold to determine the maximum number of hits to allow per trigram.

2.6.2 Index Performance

With an uncapped number of trigrams returned per hit and extending alignments (up to the length of the query) from each, the run time required per query averaged 2.6 seconds on a 1.6 GHz laptop. This was only marginally faster than the 2.7 seconds average required to perform the dynamic programming alignments with the full collection of song lyrics. The Mean Recipocral Rank (10) using this unfiltered index set dropped to 0.718, which we considered to be a rough upper-bound on the best accuracy for searching indexed queries. Misheard lyrics with correct counterparts found in the exhaustive search but missed using the index included “Bake sale! Bake sale!” for “Exhale! Exhale!” from Prodigy’s “Breathe,” and “He wears socks on the moon” for “He wakes up in the morning” from Dave Matthews Band’s “Ants Marching,” neither of which had a single trigram pair scoring above 9.0. We could have used a lower trigram score threshold but the memory required to build the full index was beyond our computing power and would likely not be feasible in a real-world application with far more songs than the 2,345 in our collection.

To further reduce the running time, we decided to limit the number of hits returned per trigram, eliminating some of the most common (and hence, least informative) trigrams in the lyrics. Since the number of correct lyrics (as well as the total number of hits) returned from the set of 146 songs mentioned above plateaued at around 60 hits per trigram (approximately 0.4 times the number of songs), we allowed a total of $0.45 \times 2345 = 1055$ hits per trigram in the full collection. Using this additional restriction, we were able to decrease the average running time per misheard lyric query to 1.2 s, less than half the time required for an exhaustive search. However, the MRR with this limitation was reduced to 0.681, about 8% less than that of the unfiltered index and 12% less than the full dynamic programming search. This result is similar to the accuracy achieved by phoneme edit distance, and would outperform standard text-based search engines in resolving misheard lyrics. While the indexing method could be improved, these results demonstrate the potential of phoneme trigram indexing to speed up misheard lyric queries for English songs.

Chapter 3

Characterizing Rhyming Style in Rap Lyrics

3.1 Introduction

In this chapter, we consider applications of probabilistic methods in identifying and characterizing rhymes in rap lyrics. Song lyrics have received relatively little attention in music information retrieval, but can provide data about song style or content that is missing from raw audio files or user-input tags. Recent work focusing on lyrics uses the meaning of lyric text words to extract song topic, theme, or mood information. Wei *et al.* [108] used sentence level clustering to generate novel keywords from centers of semantic graphs in song lyrics. Kleedorfer *et al.* [60] developed algorithms to identify the common topic in songs and allowed for music collections to be browsed by affiliated topics. Fujihara *et al.* [40] used common phrases between different songs as keywords to create a interconnected web of lyrics. This previous work tends to ignore the pattern and sound of the words themselves.

These sound features are central to rap music, providing information about vocal delivery and rhyme scheme. This data can be characteristic of different rappers, as MCs often boast of the uniqueness and superiority of their rhyming style. Mayer *et al.* [72] previously studied lyric rhymes as an aid in predicting musical genre, but this prior work ignores two stylistic features of rap lyrics: imperfect rhymes, where syllable end sounds are similar but not identical, and internal rhyme, which occurs in the middle of lines. Kawahara [59] analyzed rhyme in Japanese rap lyrics, and demonstrated that some pairs of consonants were significantly more likely to appear in imperfect rhyme than others. Krims [62] studied variations in performance and production and developed a genre system for hip hop. This included identifying categories for flow (delivery) such as speech-effusive and percussioneffusive, describing different musical styles, and naming topical themes like "mack rap" and "reality rap."

We developed a system for automatic detection of rap music rhymes to identify imperfect and internal rhymes. We trained a probabilistic scoring model of rhymes using a corpus of rap lyrics known to be rhyming, using ideas derived from bioinformatics. We then used this model to find and categorize various rhymes in different song lyrics, and assessed the model's success. High-level statistical rhyme scheme features we calculated allowed us to quantitatively model and compare rhyming styles between artists and genres. These features correlated with real-world notions of rapping style and we identified trends in their use in hip hop music over time. Finally, we used these rhyme features to classify rappers and investigated potential applications of rhyme stylometry. Our work has been presented at ISMIR 2009 [47] and 2010 [48] and is in review in the journal *Empirical Musicology Review* [50].

3.2 Background

Hip hop music is characterized by lyrics with intermittent rhymes being rhythmically chanted (rapped) to an accompanying beat. In "Old School" rap (from the late 1970s to mid 1980s), lyrics typically followed a simple pattern and contained a single rhyme falling on the fourth beat of each bar [17]. Contemporary rap features more varied delivery and many complex rhyme stylistic elements that are often overlooked [4]. Key among these are rhymes that are imperfect, extended, or internal.

3.2.1 Imperfect Rhymes

Holtman [51] provides a good overview of the abundance of imperfect rhyme (also called slant rhyme) in rap lyrics and identifies some examples of their use in Eric B. and Rakim's 1990 album *Let the Rhythm Hit 'Em* [12]. A normal rhyme involves two syllables that share the same nucleus (vowel) and coda (ending consonants). Two syllables form an imperfect rhyme if one of these two parts does not correspond exactly. However, these types of rhymes are not just composed of vowels and consonants being paired randomly: there is a constraint to the amount of dissimilarity in these rhymes, determined by the shared articulatory features of matching phonemes.

In Holtman's hierarchy, the most similar consonants are nasals, fricatives, and plosives differing only in place of articulation, as in the line-ending /m/ and /n/ phonemes in:

> Entertain and tear you out of your **frame**
> Leave you in a puddle of blood, then let it **rain**, [12]

as well as the /k/-/t/ from “black”-“fat” and /t/-/p/ from “coat”-“rope” pairs in:

> Cool, I heat you up like a **black mink coat**
> Hug your neck like a **fat gold rope**. [12]

(Rhyming syllables in quoted lyrics are displayed with the same font style.)
Less similar consonant pairs include those with the same place of articulation, but differing in voice or continuancy, such as the /k/ and /g/ pair in:

> Bring a bullet-proof vest, nothin’ to **ricochet**
> Ready to aim at the brain, now what the **trigger say**? [12]

Though vowel identity tends to be preserved in rhymes, nonidentical vowels are most similar when differing only in height or “length” (advanced tongue root), such as the penultimate vowels (/ɛ/ and /eɪ/) in:

> I’m the alpha, with no **omega**
> Beginning without the, end **so play the**, [12]

or the /ɑ/ and /ɔ/ in:

> Beats and bullets pass me, none on **target**
> They want the R hit, but watch the **god get**. [12]

Less similar vowel pairs differ in front/back position such as the /ɛ/ and /ɔ/ in:

> Vocabs is endless, vocals **exist**
> Rhyme goes on, so no one can **stop this**. [12]

Holtman’s work is largely taxonomic and describes known rhymes, rather than discovering them. Hence, we used a statistical model of phonetic similarity based on frequencies in actual rap lyrics. However, the patterns we automatically discovered largely validate her taxonomy.

3.2.2 Polysyllabic Rhymes

Rap music often features three syllable or longer rhymes with unstressed syllables following the initial stressed pair. Also known as multisyllabic rhymes or multis [17], these may span multiple words, in which case they are called mosaic rhymes. Longer rhymes can also include more than one pair of stressed syllables:

> Maybe my sense of **húmor gets ínto you**
> But girl, they can make a per**fúme from the scént of you**.[38]

(Here, the accents mark the syllables with primary stress in the six syllable rhyme.)

3.2.3 Internal Rhymes

Finally, contemporary rap music features dazzlingly complex internal rhyme. Alim [4] analyzed Pharoahe Monch's 1999 album *Internal Affairs* [77] as a case study, and identified chain rhymes, compound rhymes, and bridge rhymes. Chain rhymes are consecutive words or phrases in which each rhymes with the previous, as in:

> New York **City gritty** com**mittee pity** the fool that
> Act **shitty** in the midst of the calm the **witty**,[77]

where "city", "gritty", "committee", and "pity" participate in a chain since they all rhyme and follow each other contiguously.

Compound rhymes are formed when two pairs of line internal rhymes overlap within a single line. A good example of this is given in "Official":

> Yo, I stick around like hockey, now what the **puck**
> Cooler than **fuck**, ma*neuver* like Van*couver* Ca**nucks**,[77]

where "maneuver" and "Vancouver" are found between "fuck" and "Canucks."

Bridge rhymes are internal rhymes spanning two lines:

> **How I made it** you **salivated** over my **calibrated**
> RAPS that **validated** my ghetto credi*bility*
> *Still I be* PACKin a*gilities* <u>unseen</u>
> Forreal-a my killin a*bilities* <u>unclean</u> fa*cilities*.[77]

Here, we called pairs in which both members are internal (such as "agilities" / "abilities") bridge rhymes, and those where the first word or phrase is line-final (such as "calibrated" / "validated"), link rhymes.

3.3 A Probabilistic Model of Rhyme

As we did for misheard lyrics (detailed in Section 2.3.1), we modeled our rhyme detection program after the BLOSUM (BLOcks of amino acid SUbstition Matrix) protein homology detection algorithms [46]. In analyzing rap as opposed to misheard lyrics, we treated song lyrics as sets of sequences of syllables, with each sequence corresponding to a line of text. We used syllables instead of phonemes since they are the fundamental unit of rhyme; two similar matching sets of phonemes may not necessarily rhyme but two similar matching sets of syllables will. To determine the possibility of rhyme in candidate pairs of song lyrics, we assigned positive scores to syllables which matched with each other in known rhyming phrases more often than expected by chance, and negative scores to those which matched less often than expected by chance. We identified rhymes as those regions with syllables that, when matched to each other, had a total score surpassing a cut-off threshold.

3.3.1 A Collection of Known Rhyming Syllables

To generate models of rhyming and randomly co-occurring syllables in rap lyrics, we required a data set of known rhymes. This data set corresponded to the corpus of alignments of homologous proteins used to train the BLOSUM matrix. Our training corpus included the lyrics of 31 influential albums from the "Golden Age" of rap (1984-1994) [76, 5], chosen because they received the highest rating from *The Source*, the top-selling US rap music magazine of the time, plus nine additional albums by influential artists from the time period (Run-D.M.C., LL Cool J, The Beastie Boys, Public Enemy, Eric B. and Rakim). We downloaded lyrics from the Web and manually corrected them to fix typos and ensure that pairs of consecutive lines ended with matching rhymes, yielding 27,956 lines of lyrics (13,978 rhymed pairs), approximately 700 lines per album.

As in Section 2.3.3, we first transcribed plain text lyrics into sequences of phonemes using a wrapper we built around the Carnegie Mellon University (CMU) Pronouncing Dictionary [64], which gives phonemes and stress markings for words in North American English. To avoid the complications and computational complexity required to evaluate all possible transcriptions for heteronyms and other words with numerous pronunciations, we selected the first transcription for each word, corresponding to the most common pronunciation.

We augmented the dictionary with common elements of hip hop vernacular and slang, including terms such as "DJ," "basehead," and "AK-47," as well as a wide variety of profanity. To accommodate for variations in spelling and pronunciation in the lyrics, we implemented rules to transform pronunciations for common occurrences of these variations. For example, if a word ending with "-in" was not found in the dictionary but the same word with a final "g" added was in the dictionary, we would use the pronunciation of the found word, replacing the /ŋ/ with an /n/. This corresponded to the "-in" ending in verbs such as "runnin'," or "kickin'." If a word ending with "-a" was not found in the dictionary but the same word with the "a" replaced with an "er" was in the dictionary, we would use the pronunciation of the found word, replacing the /ɚ/ with a /ə/. This corresponded to the "-a" formation for nouns like "brotha" or "killa." We also added rules to transcribe plural endings (/z/ for "-s") and the simple future contraction "-'ll." Finally, we reduced the stress assigned to about 30 common one-syllable words of minor significance in rhyme ("a," "I," "and," etc.) to better model their actual realizations in rap performance. To handle words not found in the augmented dictionary, we added the Naval Research Laboratory's text-to-phoneme rules [32], as we did for misheard lyrics.

3.3.2 Scoring Potential Rhymes

To generate a log-odds scoring matrix for rhyming syllables, we required models for random syllables and for rhymes. For any pair of syllables i and j, the random model, $\Pr(i, j|\text{Random})$, gives the likelihood of i and j being matched together by chance while the rhyme model, $\Pr(i, j|\text{Rhyme})$, gives the likelihood of i and j being paired in a true rhyme. As in BLOSUM [46] Equation 2.1, the log-odds score was calculated as

$$M[i, j] = \ln \frac{\Pr(i, j|\text{Rhyme})}{\Pr(i, j|\text{Random})}, \tag{3.1}$$

(though we used natural logarithms instead of base 2). However, due to the extremely high number of possible syllables in the English language, creating a pair-wise syllable scoring matrix was not feasible. Instead, we reduced each syllable to its vowel (nucleus), end consonants (coda), and stress—the relevant features for determining rhyme. We approximated the coda by taking the first half (rounded up) of the consonants between adjacent pairs of vowels. As an example, given the word "example" (transcribed as /ɪ g z ˈæ m p ə l/), the codas would be /g/, /m/, and /l/. Both models were trained using the occurrence frequencies of phonemes in the training set of 40 albums mentioned above. In the random model, the likelihood of vowel a matching with vowel b is calculated by taking the product of the frequencies of a and b:

$$\Pr(i, j|\text{Random}) = \frac{F_a \times F_b}{(\sum_m F_m)^2}, \tag{3.2}$$

where F_a is the number of times phoneme a appeared in the training lyrics. The likelihoods for consonants and varying stress were calculated independently in the same manner.

For the rhyming model, the likelihood of vowels a and b being matched was calculated by taking the number of times a and b were seen matching in known rhymes, and dividing by the total number of matched vowel pairs in known rhymes. This was calculated as:

$$\Pr(i, j|\text{Rhyme}) = \frac{F_{a,b}}{\sum_m \sum_n F_{m,n}}, \tag{3.3}$$

where $F_{a,b}$ is the number of times vowels a and b appeared matched in the known rhymes. Then the log-odds score for the vowels was calculated as before:

$$\text{vowelScore}(a, b) = \ln \frac{\Pr(a, b|\text{Rhyme})}{\Pr(a, b|\text{Random})}. \tag{3.4}$$

The likelihood for consonants was more complicated since we needed to consider unmatched consonants when aligning syllable codas of differing size. For example, the following pair from Public Enemy's "Black Steel in the Hour of Chaos" has line final consonant clusters of /l d/ and /d/:

> Cold holdin' the load, the burden breakin' the mold
> I ain't lyin' denyin', because they're checkin' my code [36].

As we did when aligning misheard lyrics with actual counterparts of different lengths, we used an iterated approach to solve these problems. In the first pass over the training data, we consider rhymes in paired lines to be all syllables following the final primary-stressed syllable, after Holtman [51], and aligned consonants sequentially from left to right. For the example given, the IPA transcription ends with

the b—ur–d–e–n b–r—ea–k–i–n' the m—o–l–d
ð ə b ˈɝ d ə n b r ˈeɪ k ɪ n ð ə m ˈoʊ l d
b ə k ˈʌ z ð ɛ r tʃ ˈɛ k ɪ n m aɪ k ˈoʊ d
b–e–c–au–se they're ch–e–ck–i—n m–y c—o–de,

so the rhyme would start at the /oʊ/ vowels. The /l/ from "mold" was matched with the /d/ from "code" and the /d/ in "mold" was unmatched. Here, we introduced symbols /_*/ and /*_/ that we treated as consonants to allow for different penalties for different unmatched consonants at the beginning and end of codas. This distinction was useful since some consonants (such as the liquids /l/ and /r/) were more likely to be unmatched at the beginning of clusters, and others (often coronals, such as /d/ and /z/) were more likely to be unmatched at the ends of clusters. A simple example of this is found in the occurrences of "harm," "unarmed," and "alarmed" rhyming with "bomb" in Public Enemy's "Louder Than A Bomb." [36] In these cases, the words still form imperfect rhymes, despite the unmatched consonants.

We used these alignments to produce initial scoring matrices by calculating the statistics above. In the second pass, we identified the starts of rhymes using these preliminary matrices to score syllables in paired lines. We moved backwards from the end of the line and stopped when we encountered a negative score for a stressed syllable pair. We identified the start of the rhyme as the last positive-scored stressed syllable pair encountered. For the example above, the rhyme was identified as "breakin' the mold" with "checkin' my code."

We used the initial scoring matrices to perform global alignment[30] on matched codas to determine frequencies for consonants pairing with other consonants, and being unmatched at the start or end of the coda. When the codas for "mold" and "code" were aligned in this way, the /d/s matched with each other and the /l/ was treated as an unmatched phoneme at the start of the consonant cluster (matching with /_*/:

the b—ur–d–e–n b–r—ea–k–i–n' the m—o—l–d
ð ə b ˈɝ d ə n b r ˈeɪ k ɪ n ð ə m ˈoʊ l d
b ə k ˈʌ z ð ɛ r tʃ ˈɛ k ɪ n m aɪ k ˈoʊ _* d
b–e–c–au–se they're ch–e–ck–i—n m–y c—o—de,

These updated alignments gave us new frequency statistics from which we produced the rhyming model and log-odds scores for consonants and stress in the same way as for vowels. Finally, we normalized the consonant score by dividing by the length of the coda to avoid the problem of syllables with long codas having the consonant score dominate. Intuitively, "win" and "gin" rhyme as well as "splints" and "mints." Since all the constituent scores were log-odds, we added them together to form a combined probabilistic log score. In making this combination, we implicitly assume that all sound features are independent. This is not necessarily correct (for example, after different vowels, there are different distributions of consonants), but works well as an approximation. The final score for two given syllables i and j is the sum of the vowel score, normalized consonant score, and stress score:

$$\mathrm{Rhyme}(i,j) = \mathrm{vowelScore} i_v, j_v + \mathrm{align}(i_c, j_c) + \mathrm{stressScore}(i_{\mathrm{stress}}, j_{\mathrm{stress}}), \tag{3.5}$$

where i_v is the vowel in i, $\mathrm{align}(i_c, j_c)$ is the score for the global alignment of the end consonants of i and j, and i_{stress} is the metrical stress marking of i. Since we only considered the pronunciations, homophones and identical syllables were treated in the same manner, and both generally received high scores.

Tables 1 and 2 show the pairwise scoring matrices for stressed vowels and consonants. The symbols "_*" and "*_" indicate scores for unmatched consonants at the beginning and end of codas, respectively. The similarity of vowels differing in height only appears for back vowels: /ɑ/, /ɔ/, and /ʊ/, receiving high scores when paired. We see this less for front vowels /eɪ/, /ɛ/, /ɪ/, and /i/, though these tend not to score as negatively as other vowels when paired. In the consonant matrix, high scores for fricative pairs like (/f/,/θ/) and (/v/,/ð/), nasals (/m/,/n/), as well as plosives such as (/k/,/p/) and (/p/,/t/) largely validate Holtman's hierarchy [51]. We also see an interesting effect where affricates score highly with their constituent fricatives: (/tʃ/,/ʃ/) and (/dʒ/,/ʒ/). The consonants most likely to appear unmatched at the ends of codas include /d/, /z/, /t/, and /s/, which for the most part probably correspond to common endings for verb past tenses and noun plurals, such as "trap" rhyming with "capped" or "hot" rhyming with "rocks." (We note that "rhymes with" is not a categorical relationship in this domain: while it is not obvious if a particular listener would say "trap" rhymes with "capped," most native English speakers would likely agree that these rhyme better than "trap" with "fit.") The consonants most likely to be unmatched at the start of codas are the approximants /r/ and /l/, and /s/, as in “master” rhyming with “stature.”

	ɑ	æ	ʌ	ɔ	aʊ	aɪ	ɛ	ɝ	eɪ	ɪ	i	oʊ	ɔɪ	ʊ	u
ɑ	2.3	-3.3	-0.8	**1.6**	-1.7	-2.7	-7.2	-0.6	-3.9	-4.8	-3.9	-1.0	-1.7	-3.3	-3.9
æ		2.1	-1.5	-6.6	-1.9	-3.3	-1.5	-3.4	-1.8	-2.0	-4.3	-4.6	-4.5	-3.7	-6.7
ʌ			2.2	-1.2	-1.4	-1.4	-0.6	-0.2	-1.7	-0.3	-3.0	-1.0	-0.6	-0.9	-1.5
ɔ				3.1	-1.0	-3.8	-6.5	-1.1	-3.9	-4.2	-6.3	-0.3	-0.4	**1.1**	-3.3
aʊ					3.8	-0.3	-6.0	-4.2	-5.7	-6.0	-5.7	-2.0	-2.9	-4.5	-1.4
aɪ						2.5	-4.2	-1.1	-7.0	-1.8	-3.2	-4.3	-1.1	-5.7	-6.4
ɛ							1.9	-1.2	-1.5	**0.2**	-2.1	-7.0	-4.5	-6.1	-4.3
ɝ								3.9	-5.6	-1.5	-5.5	-1.6	-2.7	-1.3	-2.6
eɪ									2.5	-3.4	-2.7	-4.4	-4.3	-5.8	-6.5
ɪ										2.0	-0.9	-7.1	**0.2**	-2.2	-3.7
i											2.4	-4.4	-4.2	-5.8	-6.4
oʊ												2.8	-4.0	-2.5	-1.5
ɔɪ													4.9	0.1	-3.7
ʊ														2.6	-0.5
u															3.1

Table 3.1: Log-odds scoring matrix for vowels. Each value represents the natural logarithm of the ratio of the likelihood of the pair matching in a rhyme versus the likelihood of the pair matching by chance. For example, the score of 1.6 for /ɔ/ and /ɑ/ indicates that the pair is $e^{1.6}$, or approximately five, times more likely to appear matched in a rhyme than by chance. Positive scores for non-identical vowel pairs are in bold face.

For the example used in this section above, “breakin’ the mold” rhyming with “checkin’ my code,” the total score for the four syllable rhyme is 9.0: The first syllable scores -1.5 (/ˈeɪ/:/ˈɛ/) + 2.6 (/k/:/k/) + 1.0 (matched strong stress) = 2.1. The second syllable scores 2.0 (/ɪ/:/ɪ/) + 2.2 (/n/:/n/) + 0.0 (matched weak stress) = 4.2. The third syllable scores -1.4 (/ə/:/aɪ/) + 0.0 (weak stress) = -1.4. The fourth syllable scores 2.8 (/ˈoʊ/:/ˈoʊ/) + (0.4+2.3)/2 (/l d/:/_* d/) + 1.0 (strong stress) = 4.1; the sum is 9.0.

3.4 Rhyme Detection Algorithm

With our probabilistic scoring method for matched syllables in place, we needed a procedure to identify internal and end rhymes. Our technique is a variant on local alignment[30]; for each syllable, we identified its closest preceding rhyming syllable, as well its longest preceding rhyming phrase within the current and previous lines. For example, given the line

*Unob**tain**able* to the **brain** it’s *unex**plain**able* what the verse’ll do [77]

	b	tʃ	d	ð	f	g	dʒ	k	l	m	n	ŋ	p	r	s	ʃ	t	θ	v	z	ʒ	_*	*_
b	4.3	-4.8	1.1	0.4	-5.5	1.9	1.9	-6.9	-0.3	-0.5	-1.6	-5.5	0.1	-0.9	-1.6	-4.6	-1.0	-4.3	2.3	0.3	-2.5	-0.6	-1.5
tʃ		4.2	-1.6	-4.9	-0.3	0.3	0.4	1.5	-6.8	-6.6	-2.8	-5.5	1.1	-6.7	0.3	0.6	0.9	1.4	-6.1	-2.0	-2.5	-6.0	-2.6
d			2.3	-7.0	-7.6	0.1	0.2	-3.1	-1.7	-2.2	-2.2	-3.0	-1.8	-0.9	-9.0	-2.1	0.2	0.0	-0.2	0.0	-4.6	-0.2	1.2
ð				3.5	-5.6	-5.1	-4.2	-0.4	-0.2	-2.0	-7.5	-5.6	-6.2	-1.4	-7.0	-4.8	-0.3	1.3	2.8	1.1	-2.6	-6.0	-3.4
f					3.4	-1.2	-4.9	-0.3	-1.5	-1.3	-3.5	-1.6	1.1	-2.7	1.1	1.2	-0.9	4.0	0.6	-7.3	-3.2	-1.4	-2.9
g						4.2	1.9	0.0	-0.2	-1.0	-1.9	-5.7	-0.6	-0.8	-2.5	-4.9	-1.1	-4.5	0.3	-0.3	-2.7	-0.9	-2.8
dʒ							5.2	-6.3	-1.5	0.1	-0.5	-4.8	-0.2	-0.3	-0.6	0.6	-1.1	-3.6	1.4	1.0	4.1	-5.3	0.5
k								2.6	-2.9	-2.1	-2.6	-1.3	1.7	-2.1	-0.7	-0.6	0.9	0.5	-1.8	-3.1	-4.7	-1.0	-1.8
l									2.8	-1.8	-1.8	-2.8	-8.1	-0.5	-2.9	-6.6	-2.9	-6.3	-1.3	-1.6	-4.5	0.4	-1.0
m										2.7	1.8	0.7	-3.2	-1.2	-2.9	-1.1	-2.5	0.4	-0.6	-3.7	-4.2	-0.8	-1.7
n											2.2	1.2	-2.5	-1.0	-2.3	-0.7	-1.5	-0.6	-1.5	-2.1	-5.1	-0.4	-2.3
ŋ												4.1	-6.8	-2.7	-2.3	-5.3	-3.5	-5.0	-2.1	-2.0	-3.2	0.2	-3.9
p													3.3	-2.0	-1.1	-0.7	1.1	0.9	-0.6	-7.9	-3.8	-0.7	-0.8
r														2.8	-2.3	-0.8	-1.2	-6.1	-2.1	-2.2	-4.3	1.7	-0.7
s															2.6	2.4	-1.0	1.0	-2.4	0.5	0.0	0.6	0.6
ʃ																5.2	-0.6	-4.1	-1.3	-0.2	3.6	-5.8	-7.7
t																	1.7	1.6	-0.9	-9.2	-5.2	0.0	0.7
θ																		4.4	0.5	-6.1	-2.0	-5.4	-0.6
v																			2.9	-0.4	1.6	-1.2	-1.7
z																				2.6	3.0	-1.3	1.1
ʒ																					6.8	-3.7	-5.6

Table 3.2: Log-odds scoring matrix for consonants

from Pharoahe Monch's "Right Here," the middle /eɪ n/ ("ain") syllables all rhyme, while the whole of "unexplainable" also rhymes with "unobtainable."

For every line in a set of lyrics, we first constructed a two-dimensional matrix of the score for every pair of syllables in the current and preceding line. We then initialized tables for the closest and longest preceding rhymes found for each syllable. For each syllable in the current line, we moved backward in the line(s) and extended rhymes forward from entries marked as "anchors." Entries in the matrix (corresponding to pairs of syllables in the lines) were selected as anchors if they scored above a threshold and contained a stressed syllable or were line-final. We used a threshold value of 3.6, meaning anchor syllables were at least $e^{3.6}$, or about 37, times more likely to rhyme than randomly matched syllables. We considered syllable pairs to be line-final if both syllables were the same distance (of at most three syllables) from the end of their respective lines. Three syllables covers the most common types of end rhymes found in traditional rap lyrics.

From these anchor positions, we extended the rhymes forward as long as the total similarity scores were above a per-syllable threshold. We used a per-syllable threshold value of 2.7, meaning, on average, rhymed syllables were at least $e^{2.7}$, or about 15, times more likely to rhyme than randomly matched syllables. Hence, a two syllable rhyme needed to score above 5.4, a three syllable rhyme above 8.1, a four syllable rhyme above 10.8, etc. In addition to the iterative extension, we allowed a "jump"-type extension, in which one or two syllables could be skipped over if the following syllable pair formed an anchor with score above the anchor threshold. Longer polysyllabic mosaic rhymes often contain one or two syllables that do not rhyme in the midst of three or four that do. A good example of this can be found in Fabolous' "Can't Deny It":

> I **keep spittin', them clips copped on those calicos**
> **Keep shittin', with ziplocks of that Cali 'dro** [38]

where the two lines rhyme in their entirety, with the exception of "them"/"with" and "those"/"that."

After the rhyme was extended forward as far as possible, we checked to see if it was either closer than the previous closest rhyme found for the anchor syllable or longer than the longest rhyme found including that syllable. If so, we updated the "closest" and "longest" tables for each syllable participating in the rhyme, and added the rhyme to the collection for the set of lyrics. Each rhyme in the collection was stored as a pair of addresses and a length. The addresses gave the line number (from the lyrics) as well as the starting syllable for each phrase in the rhyming pair. Having only a single value for the length restricted rhymes to those without any unmatched syllables, even though this is not always the case in rap lyrics, as some syllables are deliberately missed and others under-pronounced in rhyming phrases. However, the relative rarity of these types of rhymes made it impractical to train a model for their detection.

After a set of lyrics was processed, we filtered the collection of rhymes to remove duplicates and consolidate consecutive and overlapping rhymes. For each line in the set, we compared every pair of rhymes beginning on that line. If one rhyme began exactly where another ended or began in the middle of another such that both starting syllables of the rhyme were the same distance from the starting syllables of the second rhyme, we joined the two into a single rhyme beginning at the earlier starting syllables and having the combined length of the two constituent rhymes. If one rhyme was entirely contained within another such that both of its starting syllables were the same distance from the starting syllables of the second rhyme, we removed the contained rhyme entirely. For the example given above, suppose the collection began with the following four rhymes: "keep spittin', them clips" with "keep shittin' with zip-," "clips copped on those calicos" with "ziplocks of that Cali 'dro," "shit-" with "clips," and "calicos" with "Cali 'dro." The first two rhymes would have been combined to form "keep spittin', them clips copped on those calicos" with "keep shittin', with ziplocks of that Cali 'dro," then the last rhyme (being contained entirely within the first) would have been removed from the collection.

3.5 Validating the Method

Our first test verifies that our probabilistic score for syllable rhyming is better at identifying perfect and imperfect rhymes than rules-based phonetic similarity measures. We did a 10-

fold cross validation where we chose 36 albums from the training data (of 40 albums), trained a rhyme model from those albums, and used it to score the known rhyming lines from the other four albums (true positives) as well as randomly selected lines from those four albums (presumed to be true negatives).

We developed implementations of the minimal mismatch of articulatory features and Kondrak alignment metrics to compare the performance of these scoring measures, which are based on the physical process of the human voice. In the minimal mismatch method, the similarity score for a pair of phonemes is the number of their common articulatory features (such as voicing, airflow stoppage, manner of articulation, height, length, etc.) divided by the total number of articulatory features (10). The Kondrak score for a pair of phonemes is calculated in the same way, except that the articulatory features are weighted, with the most weight being given to place and manner of articulation. The alignment method was designed for identifying pairs of words in different languages sharing a common etymological root (called cognates) [61]. We show receiver operator characteristic (ROC) curves comparing the true positive rate to false positive rate when varying the score threshold for each of the three methods in Figure 3.1. The probabilistic method significantly outperforms both simpler rules-based methods.

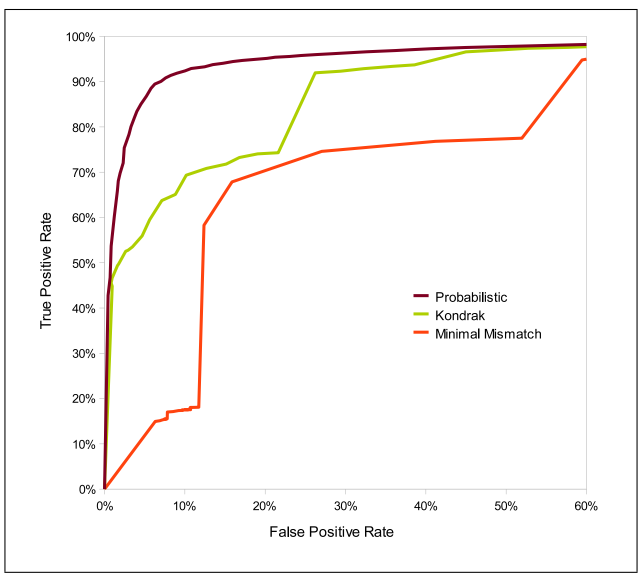

Figure 3.1: ROC curves for the three different scoring methods, comparing percentage of actual rhymes found by algorithm on the y-axis with percentage of unrelated syllables detected as rhyming on the x-axis. The kink in the Minimal Mismatch curve is caused by a sharp decline in the number of detected rhymes when zero mismatched articulatory features are allowed, meaning only perfect rhymes are counted.

Next, we considered false positives and negatives for detected end rhymes, using the score threshold of 1.5 (meaning matched syllables are at least $e^{1.5}$, or about 4.5, times more likely to rhyme than expected by chance). Out of 1000 pairs of unrelated random lines from our training data, 79 syllables were marked as parts of end rhymes ("false positives") by our procedure. Of these, 22 were in fact true rhymes, with scores higher than 3.0. Thirty were near-rhymes; that is, that they could be found (though less frequently) as line final rhymes in actual lyrics. Usually scoring above 2.0, they included matches such as "stiff"/"fit", "pen"/"thing", and "cling"/"smothering", with more than one articulatory difference or different stress. Fourteen matched end syllables (often suffixes), typically with high scores (greater than 3.0). Examples such as "weak**er**"/"drumm**er**" and "tapp**in'**"/"posit**ion**", may have exact matches, but are not relevant rhymes due to their lack of stress. The remaining 13 moderately high scoring (between 1.5 and 2.5) pairs featured either high consonant scores (like "bust"/"test") or high vowel scores due to matching rare vowel sounds ("box"/"wrong").

From a set of 1000 matched pairs of lines, we used the iterative method (moving backwards from the end of the line while scores for stressed syllables are positive) to see which true rhymes would be missed. Pairs with all such matches scoring less than 1.5 were marked and treated as false negatives. Out of 132 such syllables, the largest group (48) were moderately low scoring (between -1.0 and 1.5) pairs participating in polysyllabic and mosaic rhymes. A good example of this is "battery"/"battle me" in Eric B. and Rakim's "No Omega"[12]; many of these were flanked by high scoring pairs, and would be included in rhymes using the jump extension described in the above section. Thirty-five were very low scoring pairs (less than 0.0) which were either caused by words having been transcribed improperly or the lack of a true rhyme in the lyrics. Twenty-two were caused by the rhyme start being extended too far back and starting with a low positive scoring pair. Again, this would not cause problems in our actual detection algorithm since, in that case, rhymes are extended forward from stressed anchors. Seventeen were caused by differences between the actual pronunciation and the dictionary's pronunciation ("poems" treated as one syllable, or "battles" specifically being pronounced to rhyme with "shadows"). Finally, 10 were caused by deliberate mismatch in syllable stress.

The probabilistic model is quite good at finding both perfect and imperfect rhymes. Quite few syllable pairs (less than 15 in the 1000 line pairs) scored highly without being perceivably rhyming, and most low scoring "true" rhyme pairs take part in complex mosaic and polysyllabic rhymes.

Finally, we used our model on a set of manually annotated rap lyrics, to measure the ability of the program to find both internal and line-final rhymes. We used five songs from a variety of styles: the Beastie Boys' "Intergalactic" (1998), a Grammy-winning song in the old-school style; Pharoahe Monch's "The Truth" (featuring Common and Talib Kweli) and "Right Here" (1999), which were annotated by Alim [4] and feature high rhyme density and a complicated scheme; Jay-Z and Eminem's "Renegade" (2001), which features very high rhyme density; and Fabolous' "Trade It All (Part 2)" (2003), a song specifically mentioned by Alim for its prevalence of long (five or six syllable) rhymes. We show the ROC curves for this test set in Figure 3.2; the best overall performance is for specificity and sensitivity just above 60%. Most "false positives" were rhymes that were not annotated by the human annotator due to lack of rhythmic importance or accidental omission. False negatives included several where the performer created a rhyme from words that do not appear to rhyme as text, and some longer rhymes that were cut off prematurely due to too many non-rhyming syllables within them and lower scoring syllable pairs surrounding them. Finally, some rhymes were missed due to intervening rhymes being found between the rhyming parts, particularly when the threshold for rhymes is set low. This is especially evident in the ROC curves at lower cut-off thresholds, where true positive rates peak around 80% and begin to decline as the threshold is lowered.

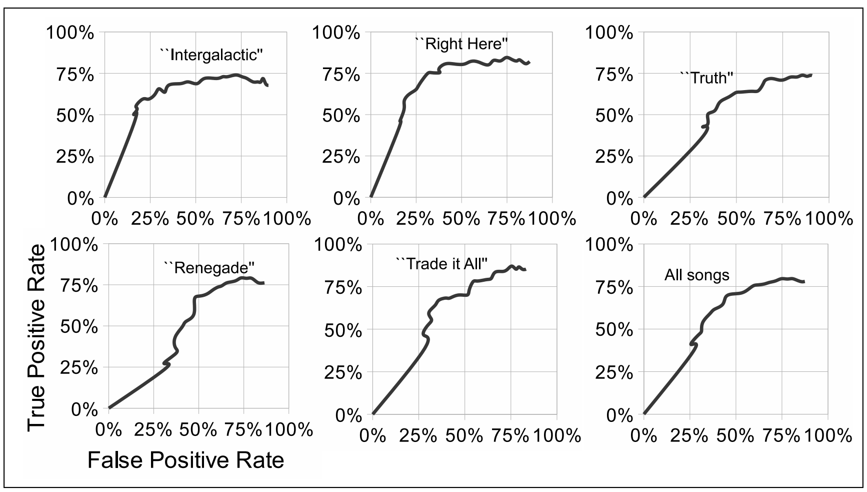

Figure 3.2: Rhyme detection syllable ROC curves for different songs. The y-axis indicates the percentage of true rhymes identified by the algorithm, while the x-axis shows the percentage of automatically identified rhymes not considered to be true rhymes.

3.6 Experiments

3.6.1 Genre Identification

We used our procedure to compute a variety of features about the rhymes in several sets of lyrics. These statistics include the number of syllables per line, the number of rhymes per line, the proportion of rhymes of different syllable lengths, as well as the occurrences of the complex rhyming features (bridge, link, chain, etc.) per line discussed earlier. The complete list of features calculated is detailed in 3.3.

Feature	Description
Syllables per Line	Average number of syllables per line
Syllables per Word	Average word length in syllables
Syllable Variation	Standard deviation of line lengths in syllables
Novel Word Proportion	Average percentage of words in the second line in a pair not appearing in the first
Rhymes per Line	Average number of detected rhymes per line
Rhymes per Syllable	Average number of detected rhymes per syllable
Rhyme Density	Total number of rhymed syllables divided by total number syllables
End Pairs per Line	Percentage of lines ending with a line-final rhyme
End Pairs Grown	Percentage of rhyming couplets in which the second line is more than 15% longer (in syllables) than the first
End Pairs Shrunk	Percentage of rhyming couplets in which the second line is more than 15% shorter (in syllables) than the first
End Pairs Even	Percentage of rhyming couplets neither grown or shrunk
Average End Score	Average similarity score of line final rhymes
Average End Syl Score	Average similarity score per syllable in line final rhymes
Singles per Rhyme	Percentage of rhymes being one syllable long
Doubles per Rhyme	Percentage of rhymes being two syllables long
Triples per Rhyme	Percentage of rhymes being three syllables long
Quads per Rhyme	Percentage of rhymes being four syllables long
Longs per Rhyme	Percentage of rhymes being longer than four syllables
Perfect Rhymes	Percentage of rhymes with identical vowels and codas
Line Internals per Line	Number of rhymes with both parts falling in the same line divided by total number of lines
Links per Line	Average number of link rhymes per line
Bridges per Line	Average number of bridge rhymes per line
Compounds per Line	Average number of compound rhymes per line
Chaining per Line	Total number of words or phrases involved in chain rhymes divided by total number of lines

Table 3.3: Description of higher-level rhyme features calculated.

We hypothesized that these features would show differences between genres of popular music, and calculated them for four sets of data: the top 10 songs from Billboard Magazine's 2008 year-end Hot Rap Singles chart; the top 20 songs from the 2008 year-end Hot Modern Rock Songs chart; the first 400 lines of Milton's "Paradise Lost" [75], as a similar-sized sample of historically important non-rhyming verse; and the top 10 songs from the 1998 year-end Hot Rap Singles chart. To compare the verses most of all, the song lyrics were modified to remove intro/outro text, repeated lines, and additional choruses. Our results are in Table 3.4. High end rhyme scores are indicative of song lyrics in general (relative to unrhymed verse); rap has higher rhyme density, internal rhyme, link rhymes, and bridge rhymes. Interestingly, blank verse and rock lyrics have similar amounts of rhyming per line, but rock lyrics have more rhymes per syllable. Although "Paradise Lost" is written in iambic pentameter (meaning it should have exactly 10 syllables per line), its use of archaic words not found in the pronouncing dictionary and shifts in English pronunciation over time have it being detected as using a bit more than 10 syllables per line. The data from 1998 and 2008 rap songs suggest that in their rhyming pattern, there has not been much shift in style, other than a possible increase in the amount of chain rhymes used.

	Rap '08	Rap '98	Rock	Blank
Number of Lines	476	613	502	400
Number of Syllables	4646	6492	4053	4146
Syllables per Line	9.76	10.59	8.07	10.37
Number of Rhymes	794	1118	476	393
Rhymes per Line	1.67	1.82	0.95	0.98
Rhymes per Syllable	0.17	0.17	0.12	0.09
Rhyme Density	0.28	0.27	0.19	0.12
Average End Score	5.28	5.21	4.36	2.49
per Syllable	3.75	3.67	4.01	2.28
Doubles per Rhyme	0.23	0.29	0.15	0.18
Triples per Rhyme	0.08	0.06	0.04	0.03
Quads per Rhyme	0.02	0.03	0.05	0.00
Longs per Rhyme	0.03	0.02	0.04	0.01
Internals per Line	0.62	0.60	0.27	0.28
Links per Line	0.20	0.28	0.13	0.16
Bridges per Line	0.43	0.48	0.28	0.40
Chaining per Line	0.32	0.18	0.15	0.07

Table 3.4: Comparison of selected rhyme features for different genres. Some of the more interesting differences are highlighted.

3.6.2 Use of Rhyme Features within Hip Hop

We also hypothesized that features of individual rappers might be informative, so we produced these statistics for popular albums by 25 famous MCs from a diverse range of styles and eras. These include some best-selling rappers [81], as well as those considered by many to be among the best of all time [78, 3]: Run-D.M.C., LL Cool J, the Beastie Boys, Rakim, KRS-One of Boogie Down Productions, Chuck D of Public Enemy, Big Daddy Kane, Slick Rick, Kool G Rap, Ice Cube, MC Hammer, Scarface, Redman, Nas, Andre 3000 of Outkast, The Notorious B.I.G., 2Pac, Bone Thugs-n-Harmony, Jay-Z, DMX, Eminem, Nelly, Fabolous, 50 Cent, and Lil' Wayne. We again focused on the rapped verses, removing any lyrics which were either spoken, sung, or performed by guest artists. Since many of the statistics involve the position of rhymes in relation to the end and middle of lines, we listened to each album to ensure that lyrics were transcribed such that each line of text corresponded to a single bar or measure of music in the song. For example, upon downloading the following lyrics from Lil' Wayne's "A Milli,"

> I do what I do an you do wat you can do about it
> Bitch I can turn a crack rock into a mountain
> Dare me
> Don't you compare me cause there ain't nobody near me
> They don't see me but they hear me
> They don't feel me but they fear me, [107]

after listening to the song, we corrected errors and transcribed them into:

> And you do what you can do about it Bitch
> I can turn a crack rock into a mountain Dare me
> Don't you compare me cause there ain't nobody near me
> They don't see me but they hear me They don't feel me but they fear me [107].

The results indicate that many of these features can be quite characteristic of different artists' styles. For example, early rappers Run-D.M.C. [97, 98], LL Cool J, [53, 54], and the Beastie Boys' [15, 16] old-school style uses less rhyme with around 1.7 rhymes per line and a rhyme density of 0.22 compared to the overall average among artists of 2.0 rhymes per line and a rhyme density of 0.27. Rakim [10, 11], recording around the same time but known for his more complex style, is detected as using more triplet rhymes (9%) than previous artists (4%). Later Golden Age rappers, such as KRS-One [89, 90], Big Daddy Kane [57, 58], and Kool G Rap [91, 92], display even higher rhyme density scores, as well as a tendency to move away from line final rhymes. This is especially the case with Chuck D [36, 37], whose 0.32 end rhymes per line is amongst the lowest in our collection and quite a bit lower than the average (0.40). Rival rappers 2Pac [1] and The Notorious B.I.G.'s [13] early albums display some fairly similar style characteristics: about two rhymes per line, 30% of rhymes being two syllables long, and 9% being longer. However, The Notorious B.I.G.'s lines are shorter in length, with, on average, 11.1 syllables compared to 2Pac's 12.4.

Faster rappers like Andre 3000 [84, 85, 86] and Bone Thugs-n-Harmony [79] can squeeze in the most syllables per line (14.7 and 17.2, compared to the average of 11.6), allowing them to achieve the most rhymes per line (3.2 and 3.6) since they rap more words that can be matched with others in adjacent lines. Andre 3000 can also be considered to be one of the most eloquent MCs, using about 1.4 syllables per word, compared to the artist average of 1.25. DMX [26, 27], on the other hand, uses the shortest words of any rapper (around 1.19 syllables), though he does tend to use more mosaic rhymes, with 17% of his rhymes being longer than two syllables. Artists from the early 2000s like Eminem [34, 35], and especially Fabolous [38, 39], also favour longer rhymes, with 19% and 30% respectively of their rhymes being longer than two syllables. The most recent MC in the group, Lil' Wayne [106, 107] manages one of the highest rhyme density scores (0.33) while using some of the shortest lines (11.1 syllables) since the early '90s. The full set of feature data by album is included in the Appendix.

These data lead us to the observation of a few key trends in the development of rhyming style in hip hop over time. Most significant is the increase in rhyme density as MCs began to use more rhymes and longer rhymes. Rhyme density and year have a moderate correlation with a Pearson r^2 of 0.23 (p-value < 0.001). This is displayed in Figure 3.3.

Low rhyme density scores at 1990 and 1991 correspond to albums by Ice Cube [21] and Scarface [82] (whose raps are more story-oriented and feature less intricate rhyming) and MC Hammer [44, 45] (who generally performed dance-style rap songs with less rhyme). The low scores at 2003 and 2005 are for 50 Cent, whose removal from the data set would result in a correlation with r^2 0.33 (p-value < 0.001). 50 Cent's more traditional and less rhyme dense style may be due to his being tutored by Run-D.M.C.'s Jam Jaster Jay, who taught him to count bars and write choruses [114]. There is another major outlier in Run-D.M.C.'s 1988 album *Tougher Than Leather*, which has the highest rhyme density score (0.43) in the entire collection. This drastic increase in the amount of rhyme is due to the duo adopting a rap style often featuring lines split by caesurae with two rhyming words appearing in each half of the line, such as the following lyrics from the title track:

> Just ***peep** *and* **keep*** but don't ***sleep** *or* **weep***
> Get ***deep** *to* **leap*** or I'll ***beep** *the* **Jeep***
> Put down the clown, get 'round the town
> I found the sound that I pound the ground [99].

In these lines, the /i p/ sounds in "peep," "keep," "sleep," etc. all rhyme, then "peep and keep" rhymes with "sleep or weep" rhymes with "deep to leap" rhymes with "beep

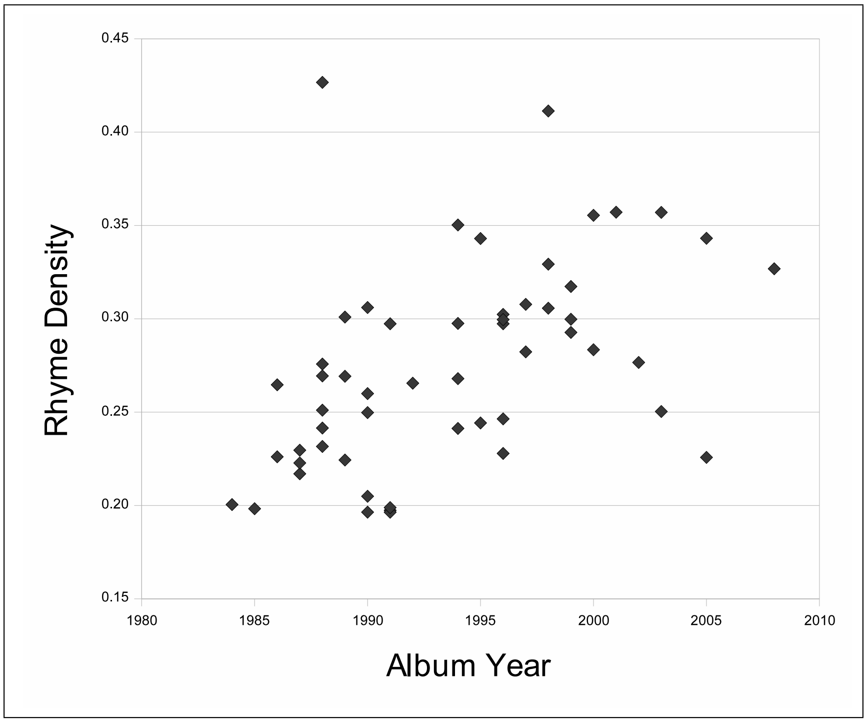

Figure 3.3: The increase in rhyme density (number of rhymed syllables divided by total number of syllables) over time. The outliers at the top left and bottom right of the graph are Run-D.M.C.'s *Tougher Than Leather* and 50 Cent's first two albums, respectively.

the Jeep," and finally "peep and keep but don't sleep or weep" rhymes with "deep to leap or I'll beep the Jeep." The same pattern occurs with the /aʊ n/ sounds in the second pair of lines. These patterns of highly repetitive rhymes result in this album having much higher rhyme density (0.43), more triplet and longer rhymes (17%), and less perfect rhymes (17%) than previous albums. However, other features less affected by this style of rhyming still appear consistent with Run-D.M.C.'s overall style: percent perfect rhymes, line-final rhyme pairs (0.46), and average score per syllable in end pairs (4.24) are all higher than the average among rappers.

The increase in longer rhymes can also be illustrated by the declining usage of one-syllable rhymes over time (see Figure 3.4). Year has a negative correlation with the percentage of rhymes being one syllable long (r^2 = 0.42, p-value < 0.001), (and matching positive correlations with the proportion of triples, quads, and longer rhymes, with r^2 all greater than or equal to 0.25, and p-values < 0.001).

Another interesting phenomenon is the increasing use of imperfect rhymes, resulting in year having negative correlations with average end syllable score ($r^2 = 0.56$, p-value < 0.001) and percent of perfect rhymes ($r^2 = 0.40$, p-value < 0.001). See 3.5 for an illustration

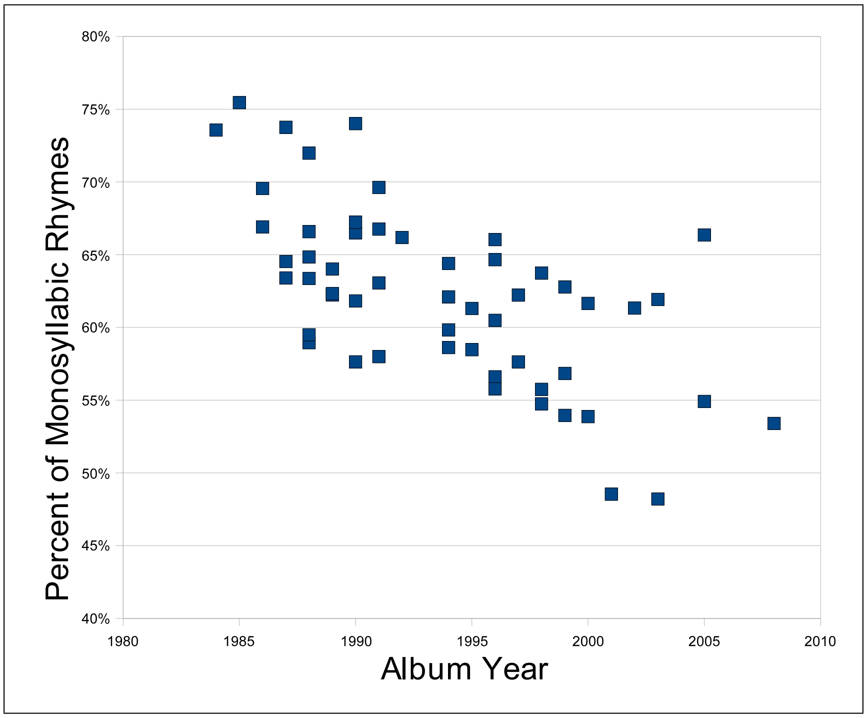

Figure 3.4: The decreasing use of monosyllabic rhymes over time.

of this relationship. Finally, we see increases in the usage of the more complex features, such as link rhymes per line, which has a correlation of 0.18 (p-value = 0.001) with year, and bridge rhymes per line, which has a correlation of 0.28 (p-value < 0.001) with year (see Figure 3.6).

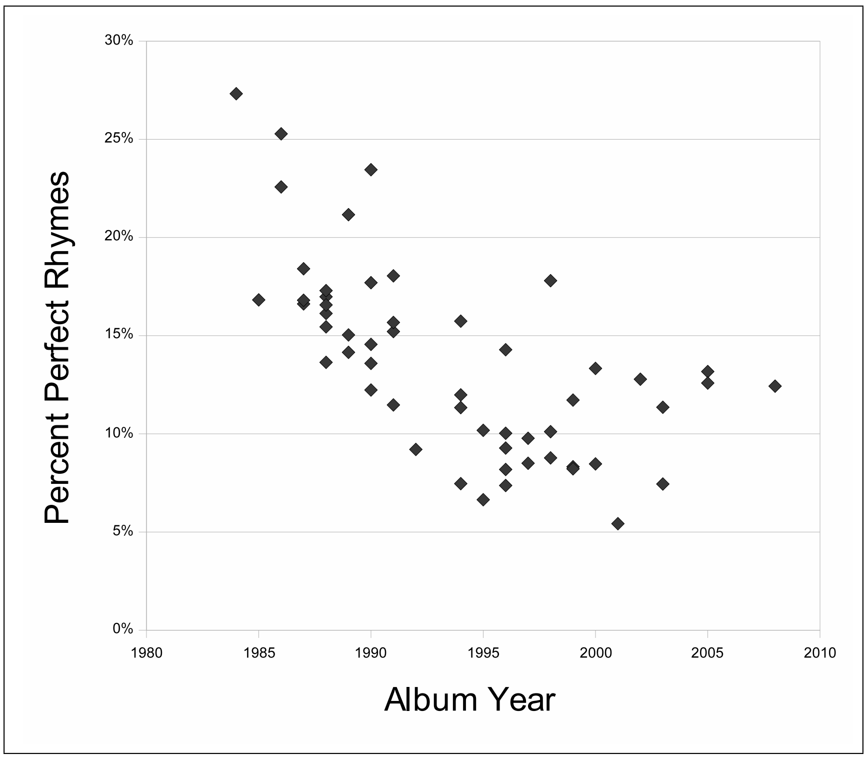

Figure 3.5: The decreasing use of perfect rhymes (in which both vowels and consonant codas match exactly) over time.

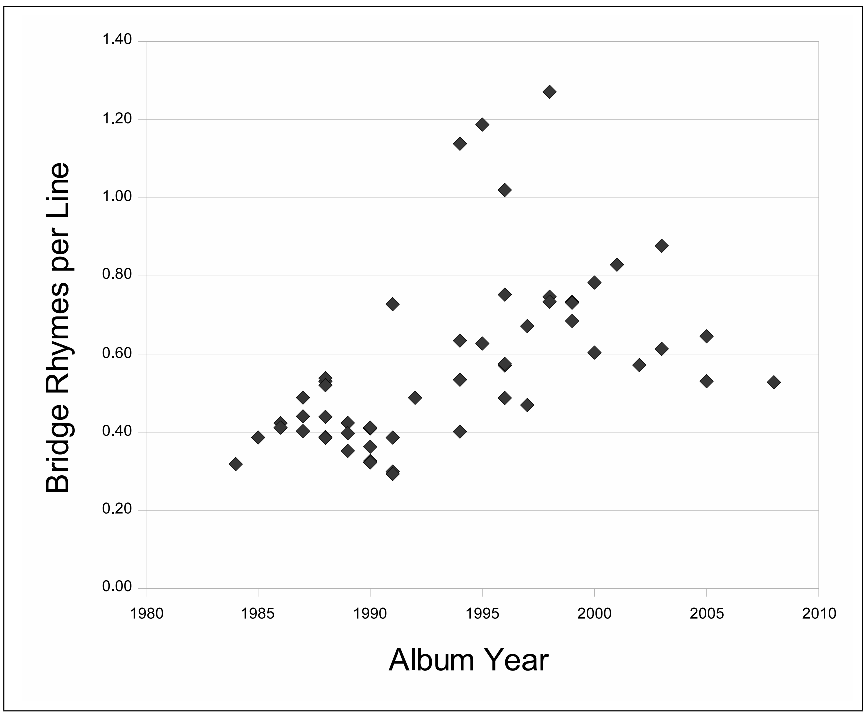

Figure 3.6: The increasing use of bridge rhymes (containing matching internal words/phrases in consecutive lines) over time.

3.7 Classifying Artists Using Rhyme Features

Most MCs in our collection displayed fairly consistent styles between albums, which often tended to be quite distinctive. We hypothesized that we might be able to classify lyrics by rapper, using only these statistical features. We broke our data set into "songs," which we treated as segments of at least 40 lines (corresponding to, at the minimum, two 16 line verses and 8 lines of chorus), and calculated the features for each of these. This resulted in 603 songs over the 53 albums. We fed this set of instances into the Weka Data Mining Software [42] and classified the data using a simple logistic regression. We used 10-fold cross-validation, in which a model was trained on 90% of the instances and used to classify the remaining 10%. The results were surprisingly good, with 314 (52%) of the instances classified correctly.

The full classification results produced a weighted F-measure of 0.516. The F-measure of a class is the harmonic mean of the precision (p) and recall (r) for that class; the precision is the percentage of instances assigned by the model to the class which actually belong to the class; the recall is the percentage of instances actually belonging to the class which are assigned to it by the model. The formula is $F_1 = \frac{2 \times p \times r}{p+r}$ [52]. All rappers were most often classified as themselves: their songs were identified as being by the correct artist more often than they were identified as being by any other particular artist, with the exception of KRS-One who was most often classified as Rakim. He, along with Jay-Z, Slick Rick, and Nelly, were among the most difficult to classify using the regression model, with F-measures all below 0.3.

Considering we used fewer than 25 fairly simple statistical features and no semantic information, the classification results are much higher than the 4% correctness we would expect by chance. A purely random classifier would achieve our level off accuracy with probability well under 10^{-50}. An obvious comparison for classification would be to use a standard bag-of-words model, which performs much better at identifying rappers. Using a naive Bayes bag-of-words classifier and the same 10-fold cross-validation as above, 552 (91.5%) instances wee classified correctly with a weighted F-measure of 0.91. However, this is not a very informative method in this genre. Rappers have a very strong inclination to name-drop in their lyrics, including their own names, nicknames, and record label and group names. This can be seen in the attributes of the naive Bayes model: the highest weighted attributes for the majority of rappers were usually one of these names. If a song has the word "jigga" in it, it is very likely to be by Jay-Z; seeing "weezy" is evidence of a Lil' Wayne song; "slim" and "shady" are indicative of Eminem; and Scarface really likes to say "Brad" (his real first name).

Furthermore, the classification errors made by a bag-of-words model tell us nothing about the style of the MCs in question. When Nelly was identified as Jay-Z, it was because his vocabulary emphasizes words like "dough," "ice," and "game," not because he rhymes like Jay-Z. Andre 3000 was often misclassified as Lil' Wayne as they share a Southern vernacular, but their rapping styles are dissimilar. Conversely, classification errors made using the rhyme features can raise very interesting comparisons of rhyming style. For example, Run-D.M.C. were often confused with the Beastie Boys. However, Run-D.M.C. had great influence upon the Beastie Boys through their association with Def Jam Recordings. In fact, Run-D.M.C. wrote or co-wrote some of the songs from the Beastie Boys' Licensed to Ill [15], including "Paul Revere" and "Slow and Low," which was originally a Run-D.M.C. recording. According to Beastie Boy Ad-Rock, "our sound right then was desperately trying to sound like Run-D.M.C." [96].

Other artists misclassified as each other included Ice Cube and Scarface, who we perceive to have a similar story-telling style, light in the use of intricate rhyming patterns, and often using uneven line lengths. The Notorious B.I.G. was most often confused with one of his influences Kool G Rap, and hip hop scholar Adam Bradley finds the similarity connection between superstars Eminem and Jay-Z to be the most interesting [18]. Table 3.5 is the full classification confusion matrix.

The rhymes we detected and the features we calculated are especially indicative of rhyming style in rap music, and they allowed us to build surprisingly useful statistical characterizations of different MCs. That these features are indicative particularly of rapping style is further supported by their relative weakness at characterizing other types of music.

We performed a similar classification experiment using ten top-selling dance/pop artists popular around the same time as the MCs in our collection: Michael Jackson, ABBA, Celine Dion, Madonna, the Backstreet Boys, Cher, Janet Jackson, Mariah Carey, Britney Spears, and Eurythmics. Just as we did for the rap lyrics, we removed repeated choruses and ensured that lines in the lyrics corresponded to bars of music in the songs. We divided the lyrics into "songs" (containing a minimum of 40 lines each), and calculated the statistical rhyme features for these songs. We used the resulting instances to train a simple logistic regression classifier and performed ten-fold cross-validation as above. The model correctly identified only 62 of 234 instances (26.5%) with an F-measure of 0.26, which is only marginally better than the 10% classification accuracy we would expect by chance.

3.8 Applications and Discussion

3.8.1 Style Modification

Given that we have a reasonably accurate statistical characterization of various rappers' rhyming styles, we can begin to consider other applications for which rhyming style can be

a	**b**	**c**	**d**	**e**	**f**	**g**	**h**	**i**	**j**	**k**	**l**	**m**	**n**	**o**	**p**	**q**	**r**	**s**	**t**	**u**	**v**	**w**	**x**	**y**	**<- classified as**
3	*6*	0	2	0	0	2	0	0	0	0	0	3	0	0	0	0	0	1	0	0	1	0	0	0	**a = KRS-One**
1	*14*	1	0	0	2	0	1	3	0	0	1	0	0	0	1	0	0	0	1	0	1	0	0	0	**b = Rakim**
0	0	*11*	*4*	0	0	1	1	0	0	3	0	0	0	0	3	0	0	2	1	0	0	0	0	0	**c = Nas**
1	1	3	*15*	0	1	0	1	0	0	0	1	0	0	0	2	0	0	1	1	1	3	0	0	0	**d = 2Pac**
0	1	0	0	*17*	0	0	0	0	1	0	0	1	0	0	0	*4*	0	0	0	0	1	0	0	1	**e = Run-D.M.C.**
0	1	0	0	0	*16*	0	0	0	1	0	0	1	2	0	1	1	2	0	2	0	3	1	0	0	**f = 50 Cent**
0	0	1	0	0	1	*17*	0	0	0	2	0	0	1	1	1	0	1	1	0	0	0	1	0	0	**g = Lil' Wayne**
1	2	0	0	0	0	0	*11*	1	1	*6*	0	0	0	0	0	0	0	0	1	0	0	0	0	0	**h = Kool G Rap**
0	*4*	0	0	1	0	0	3	*10*	0	1	2	1	0	0	1	0	0	0	0	0	0	0	0	0	**i = Big Daddy Kane**
0	0	0	0	1	0	1	1	1	*14*	2	0	1	0	0	0	0	0	0	0	0	1	0	0	1	**j = Chuck D**
1	0	2	1	0	0	2	2	1	0	*20*	0	0	0	0	2	0	0	1	0	0	1	0	0	0	**k = The Notorious B.I.G.**
0	0	0	0	0	2	0	0	1	2	0	*13*	0	*6*	0	0	2	0	0	0	0	0	0	0	1	**l = Ice Cube**
1	0	0	0	1	0	0	1	2	0	0	1	*11*	1	0	0	0	0	2	1	0	1	0	0	1	**m = LL Cool J**
0	0	0	0	0	3	1	0	0	0	0	*4*	1	*10*	0	0	0	0	0	0	0	0	1	0	0	**n = Scarface**
0	0	0	0	0	0	0	0	0	0	0	0	0	0	*23*	1	0	2	0	0	0	0	2	0	1	**o = Fabolous**
0	1	1	*5*	0	0	3	0	0	0	2	0	0	0	1	*6*	0	*5*	0	2	0	2	*4*	0	0	**p = Jay-Z**
0	1	0	0	*4*	0	0	0	2	0	0	0	1	0	0	0	*12*	0	1	0	0	0	0	0	0	**q = Beastie Boys**
0	1	3	0	0	1	3	0	0	0	0	0	0	0	0	3	0	*15*	0	0	0	0	2	0	0	**r = Eminem**
0	1	0	2	1	1	0	0	0	0	3	2	2	0	0	0	1	1	*5*	0	1	0	1	0	0	**s = Slick Rick**
0	0	2	1	0	0	0	0	0	1	0	0	0	0	0	1	0	0	1	*14*	0	0	2	0	0	**t = Redman**
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	*12*	0	0	0	0	**u = Andre 3000**
0	1	0	1	0	*4*	1	0	0	0	0	0	0	0	0	3	0	1	0	1	0	*5*	1	0	1	**v = Nelly**
0	0	0	0	0	2	1	0	0	1	1	0	0	0	2	1	0	1	1	0	0	1	*17*	0	0	**w = DMX**
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	*13*	0	**x = Bone Thugs-n-Harmony**
0	0	0	1	0	2	0	1	0	0	0	3	0	0	0	0	0	0	0	0	1	1	0	0	*10*	**y = MC Hammer**

Table 3.5: Confusion matrix for rappers using a simple logistic regression with statistical rhyme features. Each row corresponds to songs by the artist indicated at the right, while the columns display which rapper the song was identified as. The class(es) under which each rapper is most often classified is displayed in italic.

analyzed. One such application would be in inferring an artist's stylistic consistency from the accuracy by which they can be classified. For the most easily identified rappers in our collection, we can identify a single distinctive characteristic which explains their statistical uniqueness. For Andre 3000 and Bone Thugs-n-Harmony, it is their speedy flow resulting in relatively more syllables per line. For Fabolous, it is his extensive usage of extremely long multisyllabic rhymes. For other relatively well classified artists such as Run-D.M.C., Chuck D, and Redman, it is less obvious what makes their rhyming style so consistently distinct. Conversely, artists who are poorly classified can be considered to have a multitude of rhyme styles with which they can "switch up" their flow. In the song "22 Two's" from his debut album *Reasonable Doubt*, Jay-Z boasts that "I don't follow any guidelines cause too many niggas ride mine/ so I change styles every two rhymes" [55], and his resistance to classification can be seen as evidence of this constant changing of styles.

When speaking about MCs' diversity of style, we note that three of the artists in our collection are in fact not single rappers, but groups: Run-D.M.C., the Beastie Boys, and Bone Thugs-n-Harmony. However, this does not have much of an effect on the results for two main reasons. In the case of Run-D.M.C. and the Beastie Boys, many of their raps are delivered in a way such that individual members perform alternating lines (sometimes even trading parts of lines), and certain phrases (especially line-final rhymes in the case of the Beastie Boys) are performed by the whole group. This makes it extremely difficult to separate one member's contribution to the song from another's. Bone Thugs-n-Harmony generally has each of its members performing a shortened "mini-verse" of eight to twelve consecutive lines as part of longer verses. However, each of the five rappers in the group rhyme similarly enough to each other and differently enough from other rappers in the collection, that the group as a whole can be said to have a distinct style.

Rappers also sometimes consciously modify their style, perhaps in a deliberate attempt to imitate another artist's rhyme technique. A well-known example of this phenomenon occurs in the song "Notorious Thugs" from The Notorious B.I.G.'s *Life After Death* [14], which features Bone Thugs-n-Harmony as guest performers. Classifying The Notorious B.I.G.'s verse in the song using the rhyme feature logistic regression described above "incorrectly" identifies him as Bone Thugs-n-Harmony. However, his rap in this song was in fact deliberately performed so as to mimic the Bone Thugs' style. According to producer Steven "Stevie J." Jordan, "after Bone Thugs went in there and ripped it, Big took it home for a minute. He was like, 'I aint laying mine. I got to wait. This style ain't what I'm used to.'" As Bone Thugs member Layzie Bone put it, "When Biggie did our style, thats when Bone received respect for our shit" [69].

3.8.2 Ghostwriter Identification

Another interesting application of statistical rhyme style characterization is in the detection of ghostwriting in hip hop songs. While the term can refer to a range of practices in the industry, in this domain it generally refers to raps written by an artist other than the performer. We performed a small experiment on identifying ghostwriters using artists from our collection known to write for other rappers. Ice Cube was known as the primary lyricist for the seminal gangsta rap group N.W.A. and is credited with writing most of their songs. We classified Eazy-E's "Boyz-n-the-Hood" and N.W.A.'s "Express Yourself" (performed by Dr. Dre), both written by Ice Cube [31, 80], using the rhyme feature logistic regression classifier and "Boyz-n-the-Hood" was in fact identified as Ice Cube. Nas famously wrote for Will Smith on his multi-platinum solo album *Big Willie Style* [102], but using the classifier on songs "Just Cruisin"' and "Gettin' Jiggy Wit It" did not identify Nas as the rapper. However, songs written by Jay-Z for producers Dr. Dre ("Still D.R.E."), [29] and Timbaland ("Indian Carpet") [105] are both identified by the classifier as Jay-Z. As a comparison, using the naive Bayes bag-of-words classifier on these songs only correctly identifies one of them: "Boyz-n-the-Hood" is classified as Ice Cube, likely due to the inclusion of Cube-indicative words such as "fools," "hoe," and "nappy," all of which are in the top 10 attributes for his class. The results are detailed in Table 3.6.

Song Title	**Writer**	**Performer**	**Rhyme Classification**	**Bag-of-Words Classification**
Boyz-n-the-Hood	Ice Cube	Eazy-E	*Ice Cube*	*Ice Cube*
Express Yourself	Ice Cube	Dr. Dre	Big Daddy Kane	Big Daddy Kane
Just Cruisin'	Nas	Will Smith	The Notorious B.I.G	Big Daddy Kane
Gettin' Jiggy Wit It	Nas	Will Smith	Jay-Z	Jay-Z
Still D.R.E.	Jay-Z	Dr. Dre	*Jay-Z*	Fabolous
Indian Carpet	Jay-Z	Timbaland	*Jay-Z*	MC Hammer

Table 3.6: Classification results for ghostwritten songs using rhyme and bag-of-words features. Correctly identified writers are shown in italic.

The songs we used here all had the "ghostwriter" included in the credits and also include the performer as a writer, meaning that they are in fact co-written. However, for songs performed by producers (not known primarily for their rhyming abilities), we may assume that the co-writing MC had a much greater contribution to the lyrics. This may explain why the classifier does not identify Nas in the songs co-written by Nas and Will Smith, an established rapper himself. Perhaps the collaboration of the two writers resulted in a conglomerate style not characteristic of either rapper. The ability to classify Jay-Z and Ice Cube in the other cases does suggest that we may be able to identify ghostwriters even when they are not credited, which may often be the case. For example, even though rapper Skillz is perhaps best known as being a writer for other rappers [70], his ASCAP entry only has one song in which he does not rap himself [7]. (Ironically, this is Will Smith's

"Lost & Found," in which he asks, "Why should I try to flow the way ya'll flow?" [103].) Developing a statistical profile of Skillz's rhyming style might have allowed us to identify songs to which he has made uncredited contributions.

3.8.3 Content-based Recommendation

Finally, with our set of rhyme style features, we can make larger-scale comparisons between different rappers allowing for content-based recommendation in hip hop. Using normalized Euclidean distance in the 24-dimensional feature space, we built a hierarchical clustering of our albums using the Neighbour-Joining algorithm [100] (Figure 3.7). Even with this simple distance metric, the artists cluster in a reasonable way. Most artists tend to fall in small clusters with their own albums and albums by similar artists. The largest distinction is between old-school style rap (generally produced before the 1990s) and newer rap, though even within the older artists, the more intricate rhymers (Rakim and Big Daddy Kane) are branched off. Among the newer artists, there is a split between the less rhyme-dense mid-'90s (2Pac, Redman, and the Notorious B.I.G.) and other performers, who are further subdivided into the faster (Andre 3000 and Bone Thugs) and slower rappers. We can also see that the most difficult artists to classify (Jay-Z, KRS-One, Slick Rick, but surprisingly not Nelly) generally have albums that are not very similar to their other albums, indicating their diversity or progression in style.

With an embedding of MCs in a high-dimensional rhyme style space such as this one, we could easily find the two or three rappers most similar to any other artist (given enough of their lyrics to calculate the rhyme statistics). This could allow lesser-known performers to promote themselves by highlighting their similarity to more famous rappers, or let music recommendation systems make suggestions based on the rhyming styles their users prefer. Suggesting artists based on their rhyming style would be an important step towards true content-based recommendation in hip hop, where the majority of the musical information is in the lyrics.

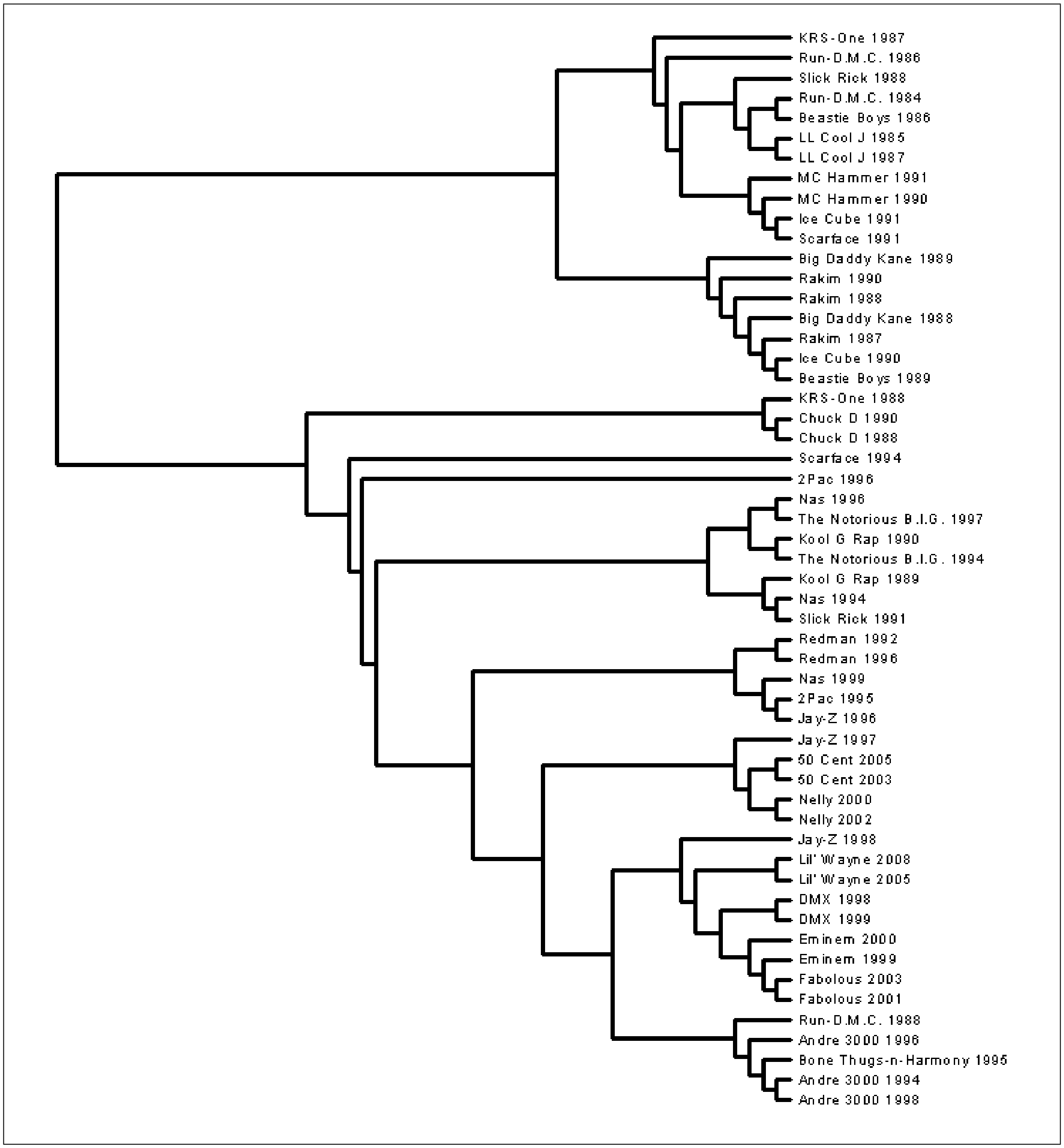

Figure 3.7: A hierarchical clustering of the albums by rhyme features built by the Neighbour-Joining algorithm.

3.8.4 Artist Evolution

To identify trends in an artist's evolving or maturing rhyme style we performed a case study on multi-platinum rapper Eminem. We calculated the statistical rhyme features for each of the seven studio albums he has released over his 15 year recording career to identify possible changes over time. The main effect we found was an overall shortening of rhyme length coupled with an increase in the amount of internal rhyme. Though only using seven albums, we found a significant negative correlation with year for percent of rhymes longer than four syllables ($r^2 = 0.80$, p-value $= 0.01$) and a significant positive correlation with year for line internals per line ($r^2 = 0.55$, p-value $= 0.05$). We also noticed trends in Eminem's decreasing use of three and four syllable rhymes, having correlations with year with $r^2 = 0.48$ (p-value $= 0.09$) and $r^2 = 0.44$ (p-value $= 0.1$), and his increasing use of single syllable rhymes and total number of rhymes per line, having correlations with year with $r^2 = 0.48$ (p-value $= 0.08$) and $r^2 = 0.53$ (p-value $= 0.07$). In fact, every single one of his albums had proportionately fewer longer than four syllable rhymes than the previous one and his poorly-received debut *Infinite*[33] had an astounding 31% of rhymes longer than two syllables. The features discussed are detailed in Table 3.7. These data lead us to believe that Eminem may have consciously modified his rhyming style to become less technically daunting and more commercially viable prior to (and during) his mainstream success.

3.8.5 Popularity Prediction

In considering the hypothesis that rhyming style could have measurable effects on popularity and critical acclaim, we compared our statistical rhyme features for each album in our collection with measures of critical reception, consumer popularity, and market success. To measure critics' reactions to the albums, we aggregated scores from professional critics' reviews compiled on the Wikipedia page for each album into a percentage grade. To measure consumer popularity, we calculated the precise average rating (out of five stars) from user reviews given to each album on the online retailer Amazon.com. Finally, to measure commercial success, we used album shipment statistics from the Record Industry Association of America [81]. These numbers are provided for albums receiving Gold, Platinum, or Multi-Platinum certifications, indicating that they have sold at least 500,000, one million, or multimillion copies. For albums not receiving RIAA certifications (BDP's *Criminal Minded* [89], Slick Rick's *The Ruler's Back* [93], and both Kool G Rap & DJ Polo albums [91, 92]), we used a conservative estimate of 100,000 units. While this may seem high, it is much closer to zero than the 500,000 minimum of a Gold album. For double albums (2Pac's *All Eyez on Me* [2] and The Notorious B.I.G.'s *Life After Death* [14]), we divided the count by two since the RIAA counts discs (as opposed to albums) in determining certifications. These data are presented in Table 3.8.

Album	Year	Rhymes per Line	Singles per Rhyme	Triples per Rhyme	Quads per Rhyme	Longs per Rhyme	Perfect Rhymes	Line Internals per Line
Infinite	1996	2.50	43%	14%	12%	5%	9%	0.91
The Slim Shady LP	1999	2.24	54%	12%	5%	3%	8%	0.78
The Marshall Mathers LP	2000	2.47	54%	10%	5%	3%	8%	0.86
The Eminem Show	2002	2.61	52%	11%	5%	3%	10%	0.98
Encore	2004	2.51	62%	7%	3%	2%	15%	1.04
Relapse	2009	2.60	55%	10%	4%	2%	10%	0.97
Recovery	2010	3.18	59%	8%	4%	1%	13%	1.27
Correlation with Year (r^2)		0.53	0.48	0.48	0.44	0.80	0.30	0.55
p-value		0.07	0.08	0.09	0.1	0.01	0.2	0.05

Table 3.7: Select features from Eminem's seven studio albums. Pearson correlation coefficient r^2 and p-values with Year are also shown.

Artist	Album	Year	Sales	Critics' score	Amazon aggregate
Run-D.M.C.	Run-D.M.C.	1984	500,000	93.6	4.82
Run-D.M.C.	Raising Hell	1986	3,000,000	91.4	4.63
Run-D.M.C.	Tougher Than Leather	1988	1,000,000	71.2	4.50
LL Cool J	Radio	1985	1,000,000	87.4	4.64
LL Cool J	Bigger and Deffer	1987	2,000,000	61.0	4.39
Beastie Boys	Licensed To Ill	1986	9,000,000	96.3	4.66
Beastie Boys	Paul's Boutique	1989	2,000,000	94.9	4.86
Rakim	Paid in Full	1987	1,000,000	94.3	4.81
Rakim	Follow the Leader	1988	500,000	90.1	4.75
Rakim	Let the Rhythm Hit 'Em	1990	500,000	79.7	4.95
KRS-One	Criminal Minded	1987	100,000	93.1	4.98
KRS-One	By All Means Necessary	1988	500,000	87.7	4.87
Chuck D	It Takes a Nation of Millions to Hold Us Back	1988	1,000,000	98.3	4.75
Chuck D	Fear of a Black Planet	1990	1,000,000	97.0	4.63
Big Daddy Kane	Long Live the Kane	1988	500,000	84.9	4.84
Big Daddy Kane	It's a Big Daddy Thing	1989	500,000	82.0	4.70
Slick Rick	The Great Adventures of Slick Rick	1988	1,000,000	82.3	4.76
Slick Rick	The Ruler's Back	1991	100,000	79.3	4.62
Kool G Rap	Road to the Riches	1989	100,000	83.0	4.14
Kool G Rap	Wanted: Dead or Alive	1990	100,000	82.5	5.00
Ice Cube	AmeriKKKa's Most Wanted	1990	1,000,000	88.6	4.35
Ice Cube	Death Certificate	1991	1,000,000	91.3	4.57
MC Hammer	Please Hammer Don't Hurt 'Em	1990	1,000,0000	50.0	3.75
MC Hammer	Too Legit to Quit	1991	3,000,000	71.0	3.67
Scarface	Mr. Scarface is Back	1991	500,000	84.0	4.28
Scarface	The Diary	1994	1,000,000	86.0	4.75
Redman	Whut? Thee Album	1992	500,000	91.1	4.73
Redman	Muddy Waters	1996	500,000	85.0	4.76
Nas	Illmatic	1994	1,000,000	93.3	4.78
Nas	It Was Written	1996	2,000,000	80.4	4.55
Nas	I Am...	1999	2,000,000	75.4	4.28
Andre 3000	Southernplayalisticadillacmuzik	1994	1,000,000	75.7	4.78
Andre 3000	ATLiens	1996	2,000,000	86.3	4.85
Andre 3000	Aquemini	1998	2,000,000	92.4	4.80
The Notorious B.I.G.	Ready to Die	1994	4,000,000	96.3	4.66
The Notorious B.I.G.	Life After Death	1997	5,000,000	90.8	4.38
2Pac	Me Against the World	1995	2,000,000	89.0	4.81
2Pac	All Eyez on Me	1996	4,500,000	88.6	4.68
Bone Thugs-n-Harmony	E. 1999 Eternal	1995	4,000,000	80.0	4.84
Jay-Z	Reasonable Doubt	1996	1,000,000	85.9	4.56
Jay-Z	In My Lifetime, Vol. 1	1997	1,000,000	73.3	4.07
Jay-Z	Vol. 2: Life and Times of Shawn Carter	1998	5,000,000	76.6	4.15
DMX	It's Dark and Hell is Hot	1998	4,000,000	74.2	4.55
DMX	...And Then There Was X	1999	5,000,000	76.5	3.77
Eminem	The Slim Shady LP	1999	4,000,000	79.8	4.21
Eminem	The Marshall Mathers LP	2000	9,000,000	93.9	4.48
Nelly	Country Grammar	2000	9,000,000	75.3	4.06
Nelly	Nellyville	2002	6,000,000	69.2	3.26
Fabolous	Ghetto Fabolous	2001	1,000,000	61.6	3.73
Fabolous	Street Dreams	2003	1,000,000	59.9	3.66
50 Cent	Get Rich or Die Tryin'	2003	6,000,000	77.3	3.46
50 Cent	The Massacre	2005	5,000,000	76.0	2.33
Lil' Wayne	Tha Carter II	2005	1,000,000	79.6	4.22
Lil' Wayne	Tha Carter III	2008	3,000,000	80.8	3.61

Table 3.8: Sales figures, aggregate critic scores, and Amazon.com user ratings for all albums in the collection

Comparing statistical features with critical reception, the largest effects we observed were for rhyme length: critics tended to favour rap with shorter rhymes as aggregate scores were negatively correlated with the percentage of four syllable ($r^2 = 0.16$, p-value $= 0.002$) and longer rhymes ($r^2 = 0.19$, p-value $= 0.001$). Smaller negative relationships were also observed with percentage of three syllable (with $r^2 = 0.08$, p-value $= 0.04$) and two syllable rhymes ($r^2 = 0.08$, p-value $= 0.04$). However, longer words were enjoyed by the critics as syllables per word had a positive correlation with aggregate score ($r^2 = 0.08$, p-value $= 0.04$). These results suggest that, when reviewing hip hop, professional music critics might be more influenced by the content of the lyrics as opposed to the intricacy and inventiveness of the rhymes.

As indicated by averaged Amazon.com ratings, music listeners also preferred shorter rhymes, though less so than the professional critics. Percentage of four syllables ($r^2 = 0.11$, p-value $= 0.01$) and longer rhymes ($r^2 = 0.13$, p-value $= 0.008$) both correlated negatively with Amazon score. Listeners also tended to pay more attention to the word content when rating music as their average scores correlated positively with syllables per word ($r^2 = 0.18$, p-value $= 0.001$) and novel word proportion ($r^2 = 0.20$, p-value $= 0.001$). Interestingly, listeners poorly rated albums with more evenly sized rhyming couplets, as the percentage of end-pairs even correlated with the number of one star (out of five) reviews ($r^2 = 0.12$, p-value $= 0.009$). This effect was mostly caused by high values for this feature (around 60%) in albums by Nelly and 50 Cent, two of the most reviled rappers in our collection. In fact, their combined four albums received more one star ratings (679) than all 50 other albums combined (531). The only other artist to receive as many negative reviews is Eminem, whose 186 one star ratings are balanced by the 1448 five star ratings his albums have received.

Comparing sales numbers with rhyme features was more difficult since overall album sales tended to rise with time (until the early 2000s); the correlation between sales and year has $r^2 = 0.12$ (p-value $= 0.009$). Though novel word proportion was negatively correlated with sales ($r^2 = 0.16$, p-value $= 0.003$), this was coupled with a larger negative correlation with year ($r^2 = 0.29$, p-value < 0.001). Similarly, average end score was negatively correlated with sales ($r^2 = 0.09$, p-value $= 0.03$), but there was also a negative trend with year ($r^2 = 0.05$, p-value $= 0.11$). The one feature where this effect was not observed was syllables per word. This statistic had a significant negative correlation with units sold ($r^2 = 0.11$, p-value $= 0.02$), meaning that albums with shorter words were purchased by more people. It is somewhat surprising that this feature was positively correlated with critics' and listeners' review scores, indicating that the most eloquent rappers receive the most critical acclaim but tend not to sell as many albums. Results like these suggest it may be possible in the future for record producers to scientifically engineer the performances on albums to maximize their commercial viability. However, the small sizes of even the most significant relationships indicate that there is far more to a rapper's critical and commercial success than we can predict with rhyme features.

3.9 User Interface

We combined the above algorithms and applications into a Java program that allows for the visualization and analysis of rhymes in rap lyrics. This easy-to-use tool can be employed by casual fans to better appreciate hip hop music, by music industry executives to compare potential rappers' rhyming styles, or even by aspiring rappers to improve their own lyrics. We used the Swing toolkit to create the graphical user interface and included five functions for processing lyric input: phonetic transcription, assigning similarity scores to lines, displaying detected rhymes, calculating rhyme features, and classifying the rapper.

3.9.1 Transcription and Scoring

The tool transcribes words from the input text using the methods described in Section 3.3.1. Due to the limitations of the output formatting, we used the CMU dictionary phonemes (as described in Table 2.1) instead of the IPA glyphs. The resulting phoneme strings are output with vertical bars (|) for word breaks and front slashes (/) for line breaks (see Figure 3.8). The tool also allows line final rhymes to be assigned log-odds similarity scores using the method and training corpus described in Section 3.3.2. Starting from the ends of consecutive lines, paired syllables are assigned scores as long as stressed pairs score above 0.

3.9.2 Displaying Rhymes

The output screen allows for the highlighting of internal and line-final rhymes using the method detailed in Section 3.4. For each rhyme detected in the lyrics, the function cycles through one of five modified formatting styles: bold face, italic, red colour, underline, and strike-through. This formatting is applied to all participating words in the rhyme pair. These formatting styles are not mutually exclusive, allowing words to be displayed as part of multiple rhymes (see Figure 3.9).

3.9.3 Analyzing and Classifying Rhyme Style

The Analyze Rhymes function produces a list of all of the statistical rhyme features (as described in Section 3.6) calculated for the input lyrics. Finally, when classifying the artist based on rhyming style, the tool loads the Weka Simple Logistic Regression model trained

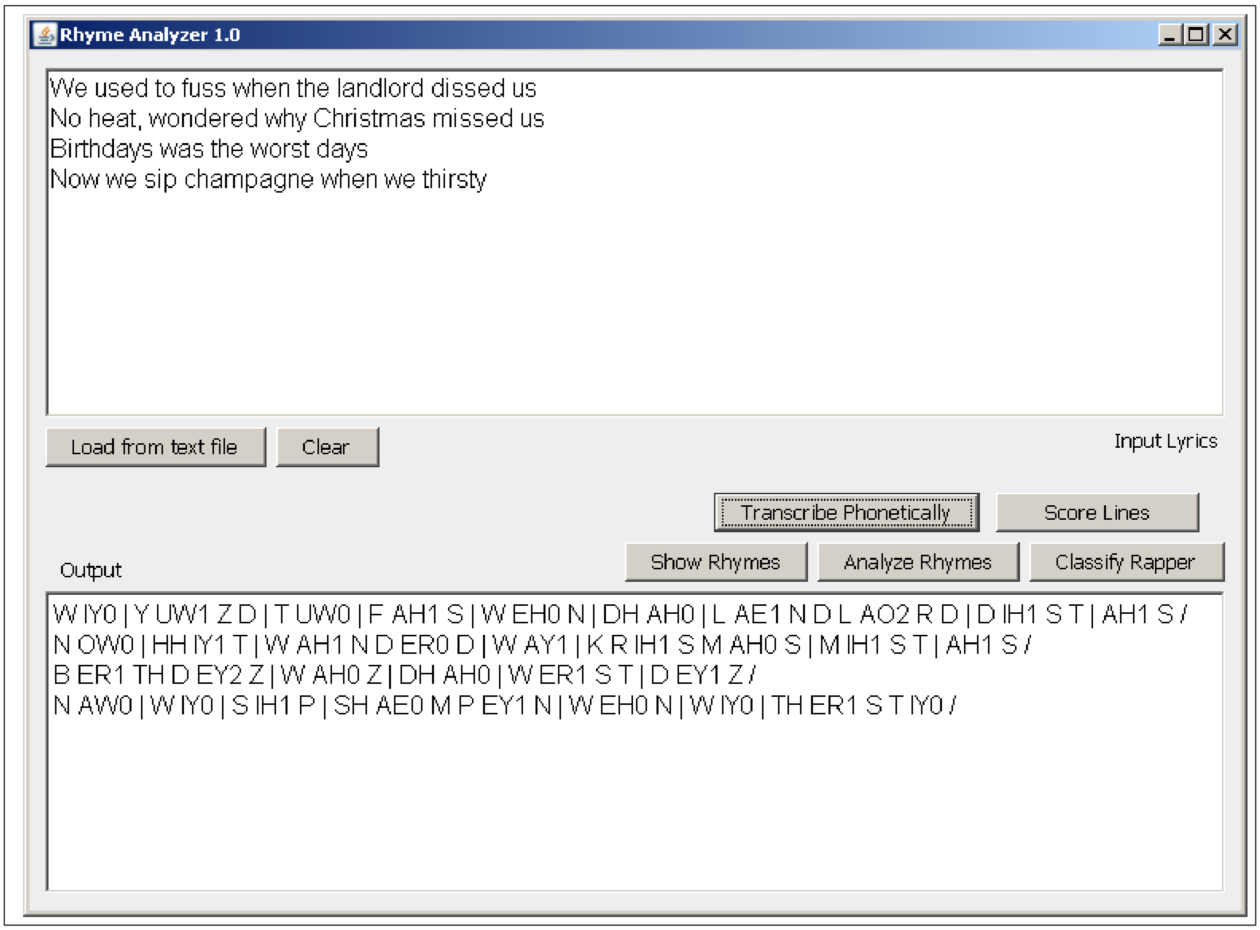

Figure 3.8: Phonetic transcription of lines from The Notorious B.I.G.'s "Juicy."

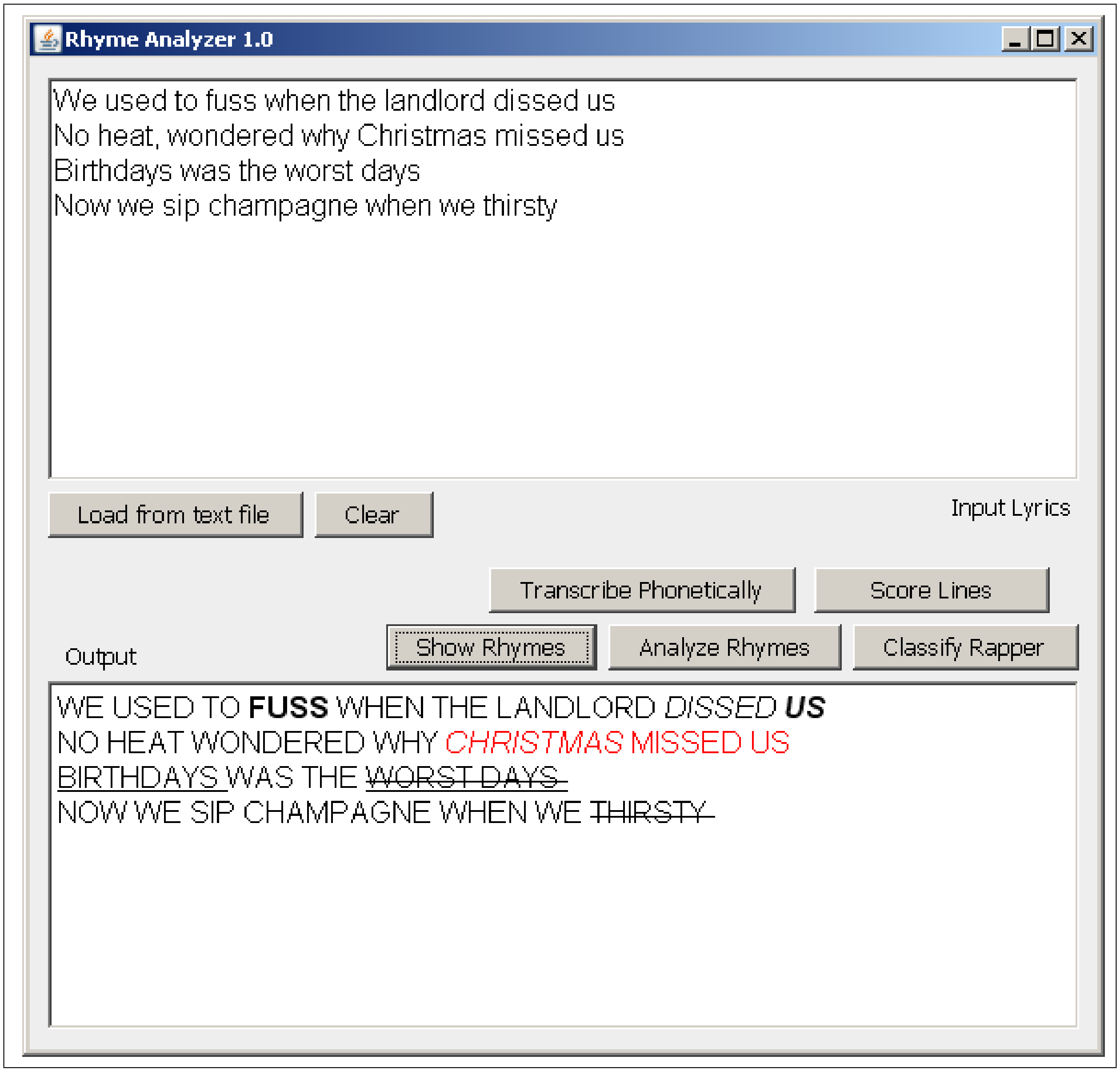

Figure 3.9: A visualization of detected rhymes from The Notorious B.I.G.'s "Juicy."

in Section 3.7. The instance generated from the rhyme features of the input text is classified by the model, and the most similar MC (from the set of 25) is returned as the guessed writer of the lyrics.

Chapter 4

Analyzing Meter in Poetry

4.1 Introduction

The final area we considered in which speech sounds could be analyzed statistically was poetry. While much of the emotional content, imagery, and wordplay is driven by the poet's choice of words and their meanings, in many poems, the prosody and rhythm of the words has an important effect on the overall feel of the piece. This is especially true for poems written in a fixed meter, in which lines of verse have regular lengths in syllables, and consistent patterns of assigned stress in those syllables. Determining the scansion of a poem (i.e., specifying this fixed meter) can be a difficult task for novice readers, so we developed an algorithm to automatically assign scansion using a dictionary of pronunciation. This also allowed for the efficient analysis and comparison of metrical style across different poets and eras. We applied this algorithm to a large corpus of poems to identify those with fixed meter, and used those poems to calculate likelihood-based stress assignments for words in the dictionary. We then investigated the use of modified prosody in metric verse and its relation to the emotionality of words used. We compared the use and form of rhyme between hip hop lyrics and poetry. Finally, we developed a software tool to perform automated scansion analysis and visualization.

4.2 Related Work

Interest in computer applications to scansion in poetry has been around for at least the past 40 years. In 1970, Donow [28] programmed an IBM 7044 to scan Shakespeare's sonnets, assign syllables to the iambic pentameter, and categorize the words based on their line position and syllable length. Dilligan and Lynn in 1973 [24] used key-punched data cards of poems and phonetic transcriptions to analyze and compare the metric complexity of works by Geoffrey Chaucer and Gerrard Manley Hopkins using the Halle-Keyser theory of prosody [43]. Logan [68] developed a set of programs to assign four levels of stress to syllables in lines of verse and measure them against an expected prosodic template to calculate a metrical complexity. More recently, Plamondon [88] developed a program to identify dominant meter and rhyme scheme in poems, using rhythmic confidence values to assign metric stress and rhyme to words not previously encountered. Our work differs from previous approaches due to our treatment of prosodic stress as a continuous likelihood of words or syllables appearing heavily stressed.

4.3 Corpus of Poems

To create a collection of poetry on which to calculate the stress likelihoods, we downloaded poems from the University of Toronto Library's Representative Poetry Online (RPO) database. Expanded in 1994 from editions of printed anthologies called Representative Poetry compiled between 1912 and 1967, the database contains 3,162 representative English poems. 600 poets are included, starting with Caedmon from the Old English seventh century, to modern poets, published in the last decade [83]. In addition to the text, entries in the RPO contain metadata about the poem, including the composition date, publication date, publication method, rhyme scheme, and form (i.e. sonnet, rhyming couplets, blank verse, etc.).

Upon downloading the html file for each poem, we removed introductory text, stanza markings, blank lines, and additional html tags, and extracted as much of the metadata as was available. As in Section 2.3.3, we used our modified CMU Pronouncing Dictionary (with reduced stress for common one-syllable words) and the Naval Research Laboratory's text-to-phoneme rules to transcribe poems phonetically. In determining prosody, we only considered the stress markings assigned to the syllables in the transcriptions and reduced these to three possible values: heavy (1) for primary and secondary stressed syllables, light (0) for unstressed syllables, and unknown (-1) for syllables transcribed using the text-to-phoneme rules. Initially, each poem was represented as a two-dimensional array of these integer values: one dimension for the lines in the poem, and the second for the syllables in each line.

4.4 Automated Scansion Algorithm

We used a fairly exhaustive searching method to determine the metrical form of a poem. For each poem, we attempted to determine the stanza size (i.e. the number of lines before the structure repeats in a regular pattern), and for each line in a stanza, the characteristics uniquely determining the meter of the line. These values are the total number of syllables, the number of light stressed syllables between heavy ones, and the number of light stressed syllables before the first heavy one. For example, a line written in iambic pentameter should have a total of 10 syllables, one light syllable between heavy syllables, and one light syllable before the first heavy one. A line written in strict dactylic hexameter should have a total of 18 syllables, two light syllables between heavy syllables, and no light syllables before the first heavy one. We specify "strict" here since the last metrical foot often does not conform exactly to the meter. This is also why we did not try to determine the number of light syllables following the last heavy syllable.

For each poem, we evaluated stanza sizes of between one and four lines. Longer stanzas are indeed common in poetry but for our purposes, allowing these tended to result in metrical structures being overfit to particular poems. Furthermore, our definition of stanza only includes the rhythmic and metrical information, not the rhyme scheme. For example, Sir John Denham's "Cooper's Hill," [23] is written in heroic couplets, meaning that it has two line stanzas of rhyming iambic pentameter. However, since every line is in iambic pentameter, we would consider it to have one line stanzas and find a single meter for all lines in the poem. For each line in a particular stanza size, if the majority of corresponding lines in the poem had the same number of syllables, we selected that value as the number for that line. The second line in a three line stanza would have corresponding lines 2, 5, 8, 11, 14, etc. in the poem.

Then, for each line position, we iterated through the possible values for the number of light syllables (one or two) between heavy ones, and the number of light syllables (zero, one, or two, but not more than the number between heavy ones) before the first heavy one. For every combination of values, we calculated a correspondence score, indicating how well the resulting metrical structure fit with the prosody assigned to the poem by the phonetic transcription. We calculated this score by performing global alignments between the stress markings for corresponding lines in the poem and a template line constructed using the characteristic metric values. For example, given a total of seven syllables, one light syllable between heavy ones, and none before the first heavy one, the template line would be [1, 0, 1, 0, 1, 0, 1]. In the dynamic programming alignment, we assigned scores of +2 to matching light or heavy syllables, +1 to matching syllables with unknown stress, 0 to mismatched stress, -3 for unmatched light syllables, -4 for unmatched heavy syllables. While selected somewhat arbitrarily, these values worked well in practice and ensured that syllables were not skipped to force poems to conform to the template meter.

For every corresponding line, we summed the resulting alignment scores and divided the total by the highest possible alignment score (+2 times the number of syllables in each line), resulting in a per line correspondence score. The average of these scores among lines was the final correspondence score for that combination of metric values, a value between 0 and

100%. After computing this result for all the iterations of values, we selected the metrical structure with the highest correspondence score. We saved these structural profiles for all poems with a correspondence score above 75%. To evaluate this cut-off, we examined 20 of the lowest scoring poems and found only one which did not match the assigned meter. This poem had eleven line stanzas in which the eleventh line in each stanza did not match the meter of the ten others. The initial set of metrical poetry contained 749 poems.

4.5 Metrical Stress Likelihood

Once we were able to identify the metrical structure with confidence for a base set of poems, we attempted to create a more robust method of assigning stress in words. While syllables in poetry do tend to be realized rather strictly as either heavily or lightly stressed, and indeed the CMU Pronouncing Dictionary does categorize syllables in this way, we predicted that many words could have variably stressed syllables. In particular, we hypothesized that different words, and especially monosyllabic words, would have measurable likelihoods of being stressed in different ways. For example, the word love is a common one syllable word in poetry usually occurring as a heavily stressed syllable but it does have some small probability of appearing as a lightly stressed syllable. By calculating the frequencies with which words appeared with various prosodic realizations in our base set of poems, we were able to identify their likelihoods of being stressed in different ways.

Restricting our corpus of poems to those with metrical structures about which we were confident, we aligned lines of poetry with their metrical template lines as described in Section 4.4. For each word, we counted the total number number of times it appeared in the poems, as well as the number of times it appeared heavily stressed. For words longer than one syllable, we counted the number of times the first syllable in the word appeared heavily stressed, as well as the number of times a syllable in the word was unmatched in an alignment. For all common words (those appearing at least ten times), we saved these values and calculated the likelihood of the word being—or having its first syllable—heavily stressed, and the likelihood of it having a skipped syllable in the meter. The 50 most common words are displayed in Table 4.1.

The first observation we made was that the most common words for the most part do not receive heavy stress. This largely validates our decision to reduce the stress of these words in the pronouncing dictionary but there do appear to be some exceptions. The words “all” and “on” are among the 20 most common though both appear heavily stressed more often than not, with “all” being stressed more than 75% of the time. “In” is the sixth most common word but it still appears heavily stressed more than 40% of the time. Perhaps unsurprisingly, “love” was the most often heavily stressed word in the top 50, appearing stressed 93% of the time. At the other end of the spectrum, we noticed an interesting effect

Word	Occurrences	Times stressed	Stress likelihood	Skipped syllables	Skip Likelihood
THE	20202	963	5%	155	1%
AND	14409	1883	13%	267	2%
TO	7972	1837	23%	79	1%
OF	7874	2209	28%	109	1%
A	6103	225	4%	30	0%
IN	5653	2341	41%	98	2%
I	4762	1238	26%	46	1%
THAT	4157	848	20%	34	1%
WITH	3788	1079	28%	59	2%
HIS	3353	265	8%	20	1%
MY	3209	218	7%	15	0%
FOR	2705	663	25%	50	2%
BUT	2554	448	18%	64	3%
ALL	2533	1928	76%	20	1%
IS	2490	812	33%	42	2%
HE	2466	600	24%	23	1%
NOT	2116	696	33%	13	1%
AS	2072	668	32%	20	1%
ON	1998	1126	56%	28	1%
FROM	1905	822	43%	18	1%
OR	1854	262	14%	16	1%
IT	1820	333	18%	26	1%
HER	1795	231	13%	9	1%
THY	1759	145	8%	4	0%
BY	1728	646	37%	24	1%
ME	1678	724	43%	16	1%
THEIR	1633	128	8%	5	0%
BE	1622	712	44%	14	1%
THEY	1589	330	21%	8	1%
AT	1458	487	33%	13	1%
NO	1458	204	14%	11	1%
WHEN	1424	593	42%	29	2%
WAS	1405	437	31%	11	1%
WE	1398	333	24%	13	1%
SO	1380	327	24%	15	1%
YOU	1290	430	33%	11	1%
THIS	1223	460	38%	9	1%
ARE	1216	358	29%	9	1%
WHAT	1171	464	40%	14	1%
THOU	1079	474	44%	14	1%
HAVE	1030	367	36%	6	1%
WHO	1012	223	22%	7	1%
WHICH	1005	346	34%	12	1%
MORE	982	642	65%	5	1%
LOVE	974	906	93%	2	0%
OUR	973	76	8%	6	1%
THEN	961	460	48%	12	1%
WHERE	960	326	34%	21	2%
YOUR	940	69	7%	3	0%
LIKE	937	397	42%	16	2%

Table 4.1: The fifty most common words and how often they appear heavily stressed or with syllables unmatched

in which the least stressed words tended to be possessive adjectives. With the exception of the extremely common “the” and “a,” the only words appearing heavily stressed less than 10% of the time were “my,” “your,” “our,” “their,” “thy,” and “his.” “Her” was the next least often heavily stressed at 13% of the time. This effect was likely due to heavy stress being applied to the more semantically important words following these adjectives and described by them.

We identified the common words most likely to appear heavily stressed (Table 4.2) and most likely to appear lightly stressed (Table 4.3), as well as those most likely to have skipped syllables (Table 4.4).

Word	**Occurrences**	**Times stressed**	**Stress likelihood**
SKIES	102	102	100%
VIEW	109	109	100%
SHADE	104	104	100%
MIND	304	302	99%
STATE	131	130	99%
DOOR	110	109	99%
SHORE	109	108	99%
RACE	107	106	99%
PAIN	201	199	99%
PLACE	270	267	99%
SIGHT	180	178	99%
BREAST	174	172	99%
NAME	214	211	99%
GROUND	120	118	98%
GRACE	172	169	98%
AGE	168	165	98%
HAIR	108	106	98%
WINGS	108	106	98%
FIELDS	107	105	98%
SOUND	137	134	98%
DIE	223	218	98%
SHOW	131	128	98%
STARS	128	125	98%
HOME	170	166	98%
CARE	207	202	98%
FACE	331	323	98%
EYE	281	274	98%

Table 4.2: Common one syllable words most likely to be heavily stressed

In comparing the most often stressed common words to the least often stressed ones, we noticed that there were far more words which rarely (less than 2% of the time) appeared lightly stressed, whereas even the least stressed word “a” appeared heavily stressed 4% of

Word	Occurrences	Times stressed	Stress likelihood
A	6103	225	4%
THE	20202	963	5%
NOR	838	54	6%
MY	3209	218	7%
YOUR	940	69	7%
WHOSE	428	32	7%
OUR	973	76	8%
THEIR	1633	128	8%
HIS	3353	265	8%
THY	1759	145	8%
ITS	641	60	9%
O	526	56	11%
AN	684	76	11%
I'LL	109	14	13%
HER	1795	231	13%
AND	14409	1883	13%
NO	1458	204	14%
OR	1854	262	14%

Table 4.3: Common one syllable words least likely to be heavily stressed

the time. This indicates that light metrical stress in words is a looser concept than heavy stress, and poets are more likely to place a normally lightly stressed syllable on a heavy beat than vice versa.

In examining the common words most likely to have syllables skipped when aligned with the metric template, we noticed that most likely of these "fire," "hour," and "power," were due to differences between the dictionary and realized pronunciations. These words have vowel triphthongs transcribed as two syllables /ˈaɪ ɝ/ and /ˈaʊ ɝ/ as opposed to their poetic usage as monosyllabic /ˈaɪ r/ and /ˈaʊ r/. The other words included "heaven," "even," and "ever," which were historically often spelled as "heav'n," "ev'n," and "ev'r," explicitly specifying their single-syllable pronunciation.

4.6 Probabilistic Prosody

With our more robust, likelihood-based metrical stress assignments, we were better able to identify the metrical structure in our collection of poems, especially for those with many one syllable words. We performed a second iteration of scansion evaluation on the poems, using the same algorithm as described in Section 4.4, though when performing local alignment between lines in the poem and metrical template lines, we used the stress

Word	Occurrences	Skipped syllables	Skip Likelihood
FIRE	224	215	96%
HOUR	141	132	94%
POWER	134	118	88%
HEAVEN	203	157	77%
MANY	288	133	46%
EVEN	157	59	38%
BEING	118	11	9%
UNTO	113	7	6%
GLORY	138	6	4%
EVER	334	14	4%

Table 4.4: Common words most likely to have unmatched syllables

likelihoods calculated above instead of hard heavy/light values. The score $c[i,j]$ for an alignment matching a particular syllable i with stress likelihood $\Pr(s_i)$ with a metrical beat j was calculated as follows:

$$c[i,j] = \begin{cases} \Pr(s_i) \times 2 & \text{(if } j \text{ is heavily stressed),} \\ (1 - \Pr(s_i)) \times 2 & \text{(if } j \text{ is lightly stressed),} \end{cases} \tag{4.1}$$

For example, "Shall I compare thee to a summer's day" from Shakespeare's Sonnet 18[101], would previously have been assigned stress as [1,0,0,1,1,0,0,1,0,1], receiving a correspondence score of 60% (= (0+0+2+2+0+0+2+2+2+2)/20) when aligned with the iambic pentameter template of [0,1,0,1,0,1,0,1,0,1]. However, using the likelihood-based assignment, the stresses would be [28%,26%,0%,100%,51%,23%,4%,100%,0%,94%], receiving a correspondence score of 76%.

With these probabilistic correspondence scores, we expanded our set of poems with metrical structures about which we were confident to 1,784 out of the full set of 3,162. We used a slightly lower cut-off correspondence of 70% since the absence of extremely rarely stressed syllables (see Table 4.3) made it much more difficult for poems to achieve the highest scores.

4.7 Modifying Meter for Poetic Affect

4.7.1 Metrical Complexity and Poetic Freedom

Using our expanded set of metrical poems and our more accurate, probabilistic measure of correspondence, we investigated poetic variations in the use of stress and meter. For poems with a fixed metrical structure, the calculated correspondence score could be seen as a measure of how strictly the poet adheres to the underlying meter. We compared these scores for poems by 22 poets from the collection, each of whom had at least 10 poems with a discovered meter. We found that Alfred Lord Tennyson kept his writing the most tightly constrained to its metrical form, as his 46 metrical poems received an average correspondence score of 79%. Sir Philip Sydney employed the most variation in his metrical structures, scoring an average correspondence under 73% in his 17 metrical poems.

While correspondence scores tended to increase slightly over time ($r^2 = 0.07$, p-value < 0.001; see Figure 4.1) as pronunciation in the English language evolved to more closely resemble the modern English of our phonetic dictionary, even poets from the same era varied in their adherence to meter. For example, Alfred, Lord Tennyson and Robert Browning both wrote in the mid 19th century, but Browning's poems received one of the lowest average correspondence scores (73.4%) and were significantly lower (p-value < 0.001) than Tennyson's, which received the highest (79.4%). Though both used similar Victorian era English, Browning's lower correspondence scores indicate his less strict reliance on metric form. Alexander Pope (born 1688) wrote before Samuel Taylor Coleridge (born 1772), but his poems received significantly higher correspondence scores (p-value $= 0.01$). Similarly, Matthew Arnold (born 1822) wrote two centuries after John Dryden (born 1631), but his poems received significantly lower correspondence scores (p-value $= 0.01$). The full list of poets, along with their years of birth and average correspondence scores is displayed in Table 4.5.

4.7.2 Stress and Emotion

We suspected that poets used words with stress patterns that did not match the underlying meter of a poem to highlight emotionally charged or figurative language. To investigate this prediction, we analyzed words in our collection of poetry using Whissell's Dictionary of Affect in Language (DAL) [109]. The DAL is a listing of over 8,700 English words along with assessor-assigned ratings (on a scale from 1 to 3) for emotional valence, emotional activation, and amount of imagery. Emotional valence is scaled from most unpleasant (1) to most pleasant (3), emotional activation is scaled from most passive (1) to most active (3), and imagery is scaled from hardest to imagine (1) to easiest to imagine (3). We augmented the dictionary with rules to allow for plurals (words ending in "-s" or "-es"), past ("-ed") and present progressive ("-ing") tenses, and noun forms ("-er"). The supplemented dictionary covered, on average, about 85% of the words in our collection of poems.

We examined DAL values for words whose stress pattern varied from the metrical stress to which they had been assigned, words likely to receive heavy stress falling on light beats

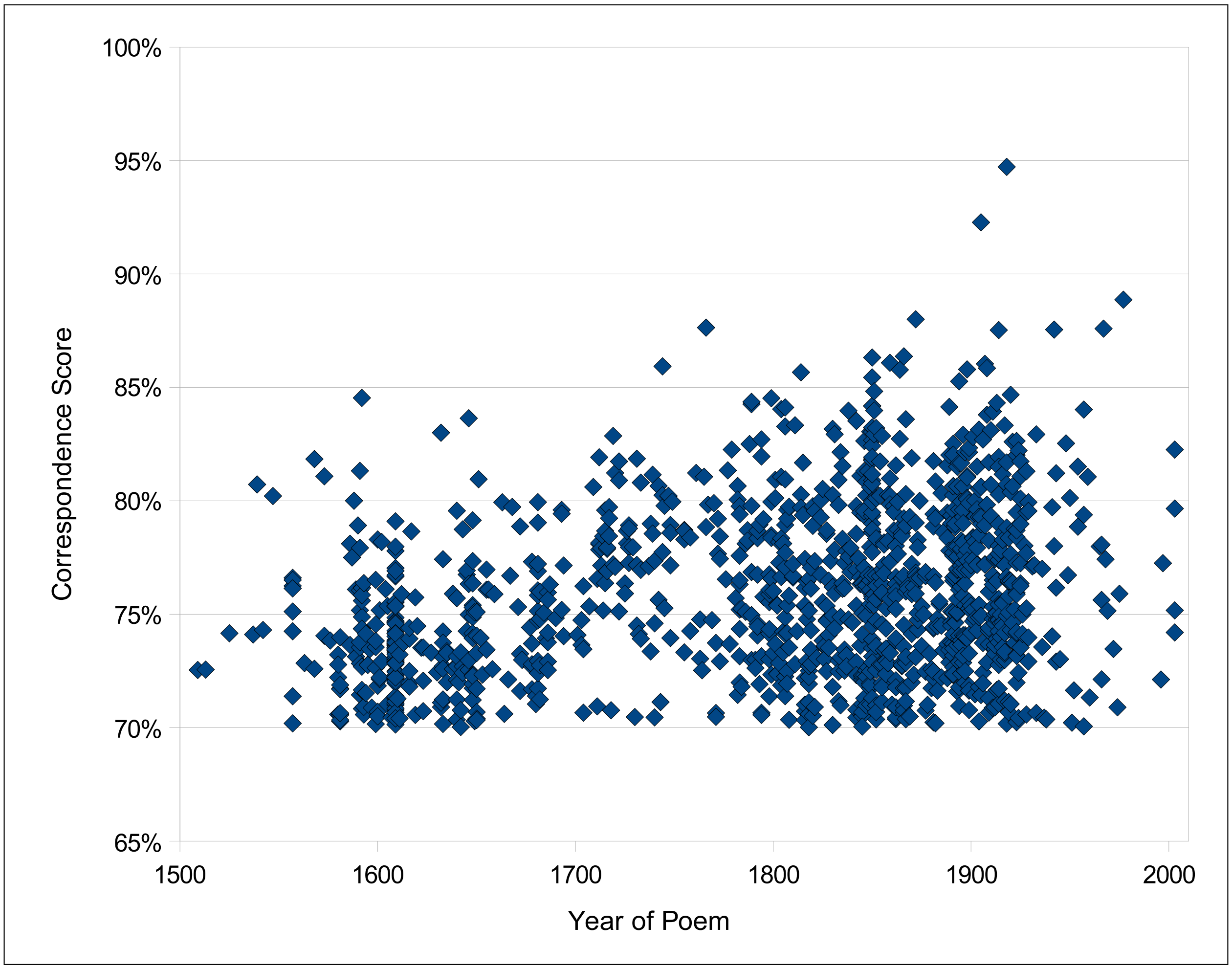

Figure 4.1: Correspondence scores by year of poem. As English pronunciation evolved to more closely resemble modern forms, dictionary-assigned syllable counts and stresses were more likely to match the poets' intended use, resulting in higher correspondence scores.

Poet	**Birth Year**	**Average Correspondence**
Sir Philip Sidney	1554	72.7%
John Milton	1608	72.9%
Matthew Arnold	1822	73.4%
Robert Browning	1812	73.4%
William Shakespeare	1564	73.4%
John Keats	1795	73.6%
Elizabeth Barrett Browning	1806	73.7%
Robert Herrick	1591	74.1%
Henry Howard, earl of Surrey	1517	74.5%
Samuel Taylor Coleridge	1772	74.6%
George Meredith	1828	74.7%
William Cowper	1731	75.1%
Andrew Marvell	1621	75.2%
Archibald Lampman	1861	75.6%
Henry Wadsworth Longfellow	1807	75.8%
Charles Tennyson Turner	1808	75.9%
William Wordsworth	1770	76.1%
William Blake	1757	76.3%
Robert Burns	1759	76.3%
John Dryden	1631	76.4%
Emily Dickinson	1830	77.0%
Alexander Pope	1688	77.4%
Walter Savage Landor	1775	77.5%
Rudyard Kipling	1865	77.8%
Henry Lawson	1867	78.0%
Alfred Lord Tennyson	1809	79.4%

Table 4.5: List of poets with 10 or more metrical poems, along with birth year and average correspondence score

(up-stressed), and compared them to the background values for all the words in a poem. We used a threshold value of 75% to determine which words were assigned to unlikely beats, so any syllable with a stress likelihood higher (lower) than 75% (25%) falling on a light (heavy) beat was considered to have varied stress. For example, "my," with stress likelihood of 7%, falling on a heavy beat was considered to have varied stress; "fire," with stress likelihood of 96%, falling on a light beat was considered to be up-stressed (and also varied in stress.)

Focusing on the mean emotional valence, activation, and imagery values for varied stress, up-stressed, and all words in each poem, we found that varied stressed words had significantly higher (p-value < 0.001) emotional activation values (mean $= 1.71$) than the background values (mean $= 1.66$) for all words. For up-stressed words in particular, mean emotional valence (1.95), emotional activation (1.83), and imagery (1.94) were all higher than the background means (1.87, 1.66, and 1.60) with p-values all less than 0.001. Since the most evocative emotional words tend to fall at the extremes of valence and activation scores, we compared the standard deviations of the values as well. We found that the valence and activation values for up-stressed words varied significantly more (standard deviations $= 0.45$ and 0.39, F-distribution p-values < 0.001) than average (standard deviations $= 0.39$ and 0.36, respectively).

As a supplement to the DAL, Whissell provides a categorization of words into Very Pleasant (within the top 10% of valence ratings), Very Unpleasant (within the bottom 10% of valence ratings), Very Active (top 10% activation), Very Passive (bottom 10% activation), Well Imaged (top 10% imagery), Poorly Imaged (bottom 10% imagery), Fun (top 25% valence and activation), Very Sad (bottom 10% valence and activation), Nasty (bottom 25% valence, top 25% activation), and Nice (top 25% valence, bottom 25% activation) words. We compared the occurrence of these words among varied stress, up-stressed, and normally stressed words. While words with varied stress tended to have similar frequencies of these special words, up-stressed words were more likely to be Very Pleasant (13%) or Unpleasant (9%) than normal (8% and 5%). They were also more likely to be Very Active (9% vs. 5%) and less likely to be Very Passive (13% vs. 22%). They were more likely to be Well Imaged (17% vs. 10%) and much less likely to be Poorly Imaged (9% vs. 41%). Finally, they were more likely to be Fun words (11% vs. 6%), but not much more likely to be Very Sad, Nasty, or Nice. These results indicate that poets tend to use words with strong stresses on weak beats to highlight the most emotionally active and figurative language in their works.

4.8 Rhyme in Poetry

Although the RPO poems are not all in rhyming couplets like our corpus of rap lyrics, we were able to use the rhyme scheme information provided in the metadata to train a probabilistic model of rhyme in poetry. This allowed us to compare the formation and definition of rhyme between hip hop and traditional poetry. For all poems with a simply parsable rhyme scheme in their metadata, we followed the method described in Section 3.3.2 to produce pairwise phoneme log-odds scoring matrices from the resulting 14,244 rhyming pairs. We first trained a model using all syllables following (and including) the last heavy syllable in a pair of lines, then used these initial scoring matrices to train a final model based on positive scoring line-final syllable pairs. The scoring matrices for consonants and vowels and in heavy syllables are displayed in Tables 4.6 and 4.7.

	b	tʃ	d	ð	f	g	dʒ	k	l	m	n	ŋ	p	r	s	ʃ	t	θ	v	z	ʒ	_*	*_
b	4.3	-4.5	-6.3	-3.7	-4.7	-3.5	-3.4	-0.8	-6.0	-5.5	-2.0	-4.9	-4.8	-6.4	-1.5	-3.8	-6.6	-4.5	-5.4	-1.5	-2.2	-0.8	-1.5
tʃ		3.0	-7.4	-4.7	-5.7	-4.5	-4.5	-6.4	-7.0	-6.6	-1.9	-1.3	-5.8	-2.8	-1.9	-4.8	-1.9	-5.6	-6.5	-1.9	-3.2	-6.4	-2.2
d			2.9	-6.6	-7.6	-1.8	-1.7	-2.3	-1.9	-1.9	-1.6	-1.1	-1.5	-1.4	-2.4	-0.3	-0.6	-2.1	-1.4	-1.8	-5.1	-1.5	-0.5
ð				4.1	-5.0	-3.8	-3.7	-5.7	-6.3	-5.8	-1.6	-5.1	-5.1	-6.6	-6.4	-4.0	-2.2	2.1	-5.7	0.1	-2.5	-5.6	-7.4
f					3.4	-4.8	-4.7	-2.0	-1.6	-2.2	-1.9	-1.5	-6.1	-1.9	-1.7	-5.0	-7.8	-5.8	-6.7	-7.4	-3.4	-0.6	-3.1
g						4.7	-3.5	0.2	-1.5	-5.6	-1.0	-4.9	-4.9	-1.8	-6.2	-3.8	-0.1	-4.6	-5.5	-6.2	-2.3	1.8	-2.6
dʒ							5.4	-5.4	-6.0	-0.3	-0.9	-4.9	-4.8	-1.1	-6.1	-3.8	-6.6	0.1	-0.8	-0.9	-2.2	-5.4	-1.9
k								3.3	-2.3	-1.3	-1.9	-1.5	-1.5	-1.9	-1.6	-5.7	-8.6	-1.9	-1.7	-8.1	-4.2	-0.2	-1.0
l									2.9	-1.9	-1.3	-1.2	-1.4	-1.2	-1.2	-6.3	-2.5	-0.7	-8.0	-2.7	-4.8	-0.6	-2.4
m										3.3	-0.7	-1.0	-1.6	-1.3	-2.3	-5.9	-1.8	-6.7	-1.3	-1.9	-4.3	-1.8	-1.7
n											2.6	-1.0	-2.0	-1.2	-0.9	-1.6	-1.6	-1.0	-1.0	-1.7	-5.4	-0.6	-1.4
ŋ												4.3	-6.2	0.0	-1.0	-5.2	-1.8	-6.0	-6.9	-1.2	-3.6	-1.1	-1.9
p													3.9	-1.5	-2.9	-5.1	-1.0	-1.3	-1.1	-2.9	-3.6	-1.0	-2.5
r														2.6	-1.6	-6.7	-2.0	-0.6	-1.8	-1.9	-5.1	1.0	-0.7
s															2.8	-1.2	-2.4	-7.2	-1.2	0.6	-4.9	-0.4	-1.4
ʃ																5.0	-6.9	-4.9	-1.1	-1.9	2.1	-1.1	-2.9
t																	2.4	-1.3	-2.9	-1.9	-5.3	-1.4	-0.8
θ																		4.3	-6.5	-7.3	1.3	-1.9	-1.2
v																			3.2	-8.2	-4.2	-1.7	-2.0
z																				2.8	-4.9	-2.8	-0.8
ʒ																					6.3	-4.1	-5.9

Table 4.6: Pairwise log-odds scoring matrix for consonants trained using rhymes from traditional poetry

The first main difference we observed was the relative lack of imperfect rhymes for consonants in the poetry. Hip hop features a wide variety of non-identical but similar consonants pairs being matched in rhymes, including voiceless stops like /k/, /p/, and /t/, the nasals /m/, /n/, and /ŋ/, and fricatives like (/f/,/θ/) and (/ʃ/,/ʒ/), which all received positive scores when paired. Conversely, in poetic rhyme, only one of these pairs (/ʃ/,/ʒ/) received a positive score. In fact, while (/k/,/t/) is a fairly common rhyming consonant pair in hip hop, it received one of the lowest log-odds scores (-8.6) in the scoring matrix trained using rhymes in traditional poems.

	ɑ	æ	ʌ	ɔ	aʊ	aɪ	ɛ	ɝ	eɪ	ɪ	i	oʊ	ɔɪ	ʊ	u
ɑ	1.88	-1.34	-0.41	1.09	-2.74	-2.17	-1.29	-0.04	-2.14	-1.83	-2.48	-0.23	-4.12	-1.37	-1.29
æ		1.57	-1.17	-1.8	-3.73	-2.51	-1.66	-2.47	-0.53	-2.81	-1.6	-3.01	-4.46	-5.65	-2.72
ʌ			1.88	-0.04	-1.03	-1.4	-0.15	-1.21	-0.57	-1.31	-1.3	0.27	-1.12	0.51	0.75
ɔ				2.54	-1.41	-1.62	-1.42	0.82	-1.2	-1.72	-1.68	0.09	-1.61	-0.24	-2.54
aʊ					3.21	-1.34	-1.67	-0.03	-1.23	-1.84	-0.92	0.74	-0.98	-1.52	-0.55
aɪ						2.48	-1.49	0.55	-1.31	-0.14	0.6	-1.86	0.78	-2.61	-2.33
ɛ							1.76	-0.13	0.06	-0.13	-0.51	-2.67	-1.76	-2.05	-1.57
ɝ								3.07	-0.66	-0.49	-1.07	-3.03	-3.16	-0.63	-1.69
eɪ									2.75	-1.5	-0.41	-1.37	-4.35	-2.11	-1.81
ɪ										1.71	-0.37	-2.57	-2.17	-3.36	-1.46
i											2.49	-2.2	-1.29	-2.09	-2.87
oʊ												2.78	-0.55	-2.12	-0.46
ɔɪ													5	-2.9	-3.68
ʊ														2.77	0.9
u															2.85

Table 4.7: Pairwise log-odds scoring matrix for heavily stressed vowels trained using rhymes from traditional poetry

Another interesting difference we saw was the extent of variation allowed in matching vowels in poetic rhyme, compared to their relative conservation in hip hop rhyme. This can be seen in the high scores for pairs like (/aɪ/,/i/), (/aʊ/,/oʊ/), (/ʌ/,/oʊ/), and (/ʌ/,/u/). While some of these may have been due to changes in English pronunciation over time, it seemed that many of these pairs were not even perceivably close rhymes. However, upon closer examination of the rhyming lines, we discovered the cause of these surprising pairings: eye rhymes. These are pairs of words which end with the same spelling, so that they "look like" rhymes. These rhymes can explain high scores for pairs like (/ʌ/,/u/) as in "All things by thee are measur'd; thou by none" rhyming with "All are in thee thou in thyself alone" from the anonymous 1602 poem "Eternal Time, that Wastest Without Waste," and (/aʊ/,/oʊ/) as in "There is no effort on my brow" rhyming with "I rush with the swift spheres and glow" from Matthew Arnold's 1852 poem "Morality."[8]. The most common word participating in eye rhymes was "love," which in Aphra Behn's 1680 poem "The Disappointment"[111] for example, rhymes with both "strove" (/ʌ/,/oʊ/) and "improve" (/ʌ/,/u/). The modeling of rhyme allowed us to identify eye rhymes as a feature unique to poetry since it is often written to be read, as opposed to hip hop lyrics which, when written, are meant to be rapped and heard.

4.9 Scansion Analysis Tool

Similar to the rhyme analysis tool described in Section 3.9, we created a Java program that allows for the identification and display of meter in poetry. This could be used as both an analysis tool, and as an aid for readers of poetry having difficulty in determining the scansion of a particular poem. We used the Swing toolkit for the graphical user interface and included three functions for processing input poetry: phonetic transcription, analyzing the meter of the poem, and showing the detected scansion.

4.9.1 Transcription and Analysis

Transcription is performed as described in Section 3.9.1. The meter of the input poem is identified using the stress likelihood-based method described in Section 4.6. The analysis output includes the stanza, and for each line in the stanza, an English description of the identified meter. If there is one weak stress between heavy ones, the meter type is iambic or trochaic (depending on the weight of the first syllable). If there are two weak stresses between heavy ones, the meter type is dactylic or anapestic (depending on the weight of the first syllables). The metric length is calculated by dividing (rounding down) the total number of syllables per line by the size of the metric foot (one plus the number of weak stresses between heavy ones). See Figures 4.2 and 4.3 for examples.

4.9.2 Displaying Scansion

The detected scansion can be displayed on the output screen using bold face font for heavily stressed syllables and red colour for those which do not match the meter of the line. We used a threshold value of 30% for this distinction, meaning that any syllable with a calculated stress likelihood less [greater] than 30% [70%] falling on a heavy [weak] beat would be coloured red. Extra syllables which do not fall on a metrical beat are displayed in strikethrough format (see Figure 4.4).

Since our pronouncing dictionary only provided syllable counts for the phonetic transcriptions of words, we required a means of parsing syllables from the plain text words in the output. We first segmented the words into chunks with a single vowel and split the consonants between vowels. While the number of chunks was less than the number of required syllables, we tried to find and split first clusters containing a consonant or apostrophe followed by a "l," "m," or "r;" then any pair of consecutive consonants; then any pair of consecutive letters. While the number of chunks was greater than the number of required syllables, we combined first chunks containing a "y;" then those ending with an "e;" then any pair of consecutive vowels; then any pair of consecutive chunks.

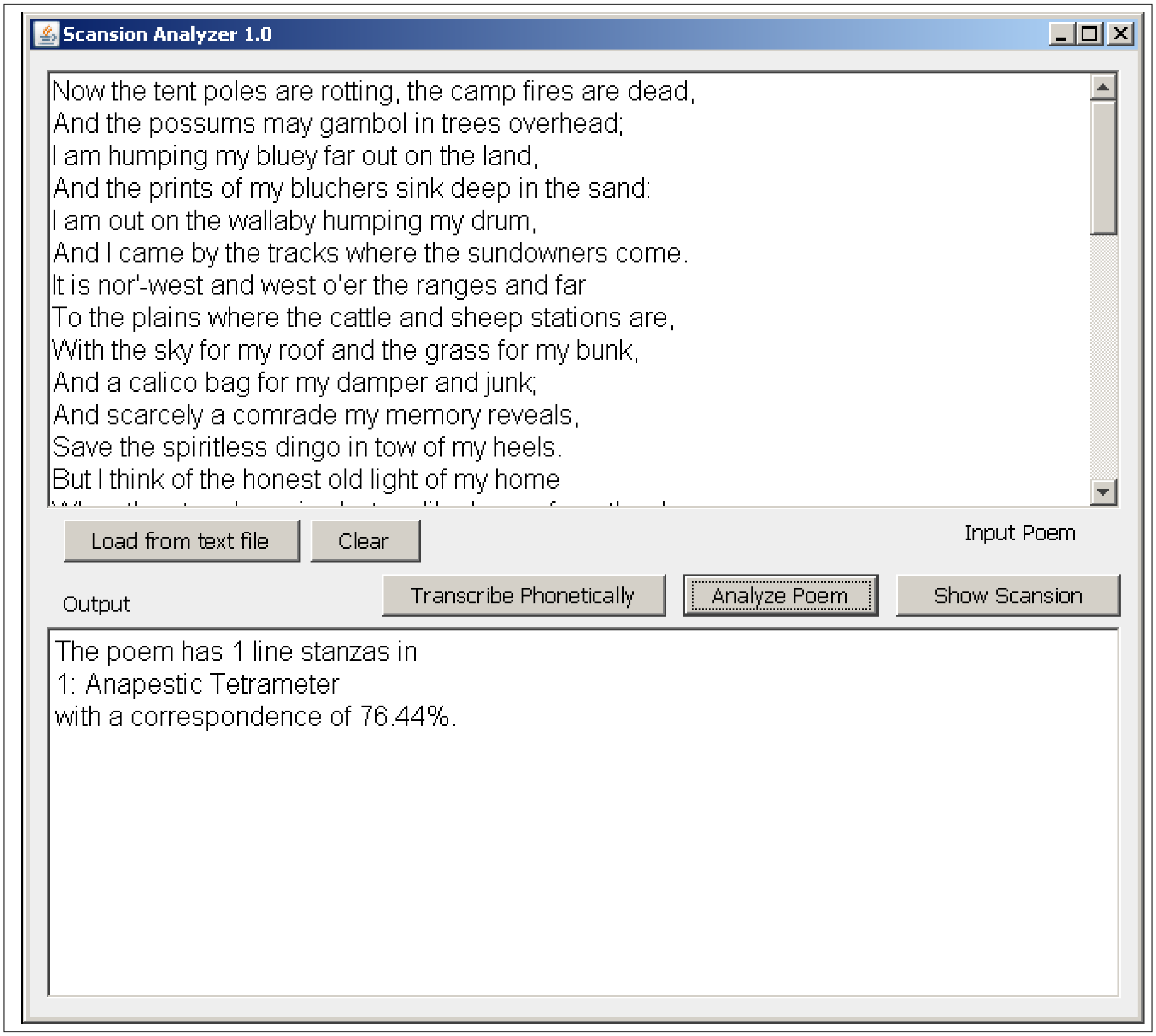

Figure 4.2: Metrical analysis of Henry Lawson's 1891 poem "Freedom on the Wallaby" [63]

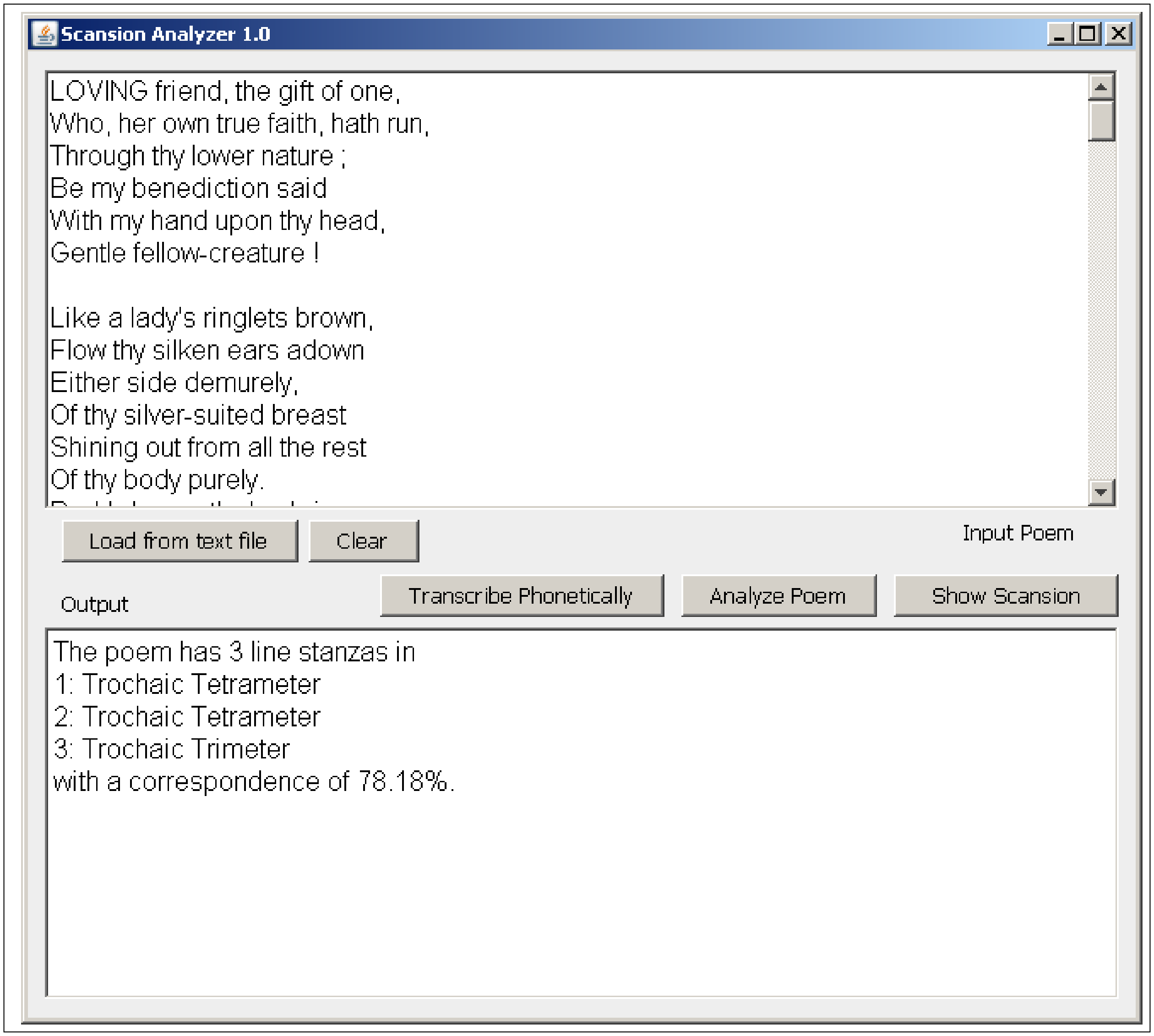

Figure 4.3: Metrical analysis of Elizabeth Barrett Browning's 1844 poem "To Flush, My Dog" [19]

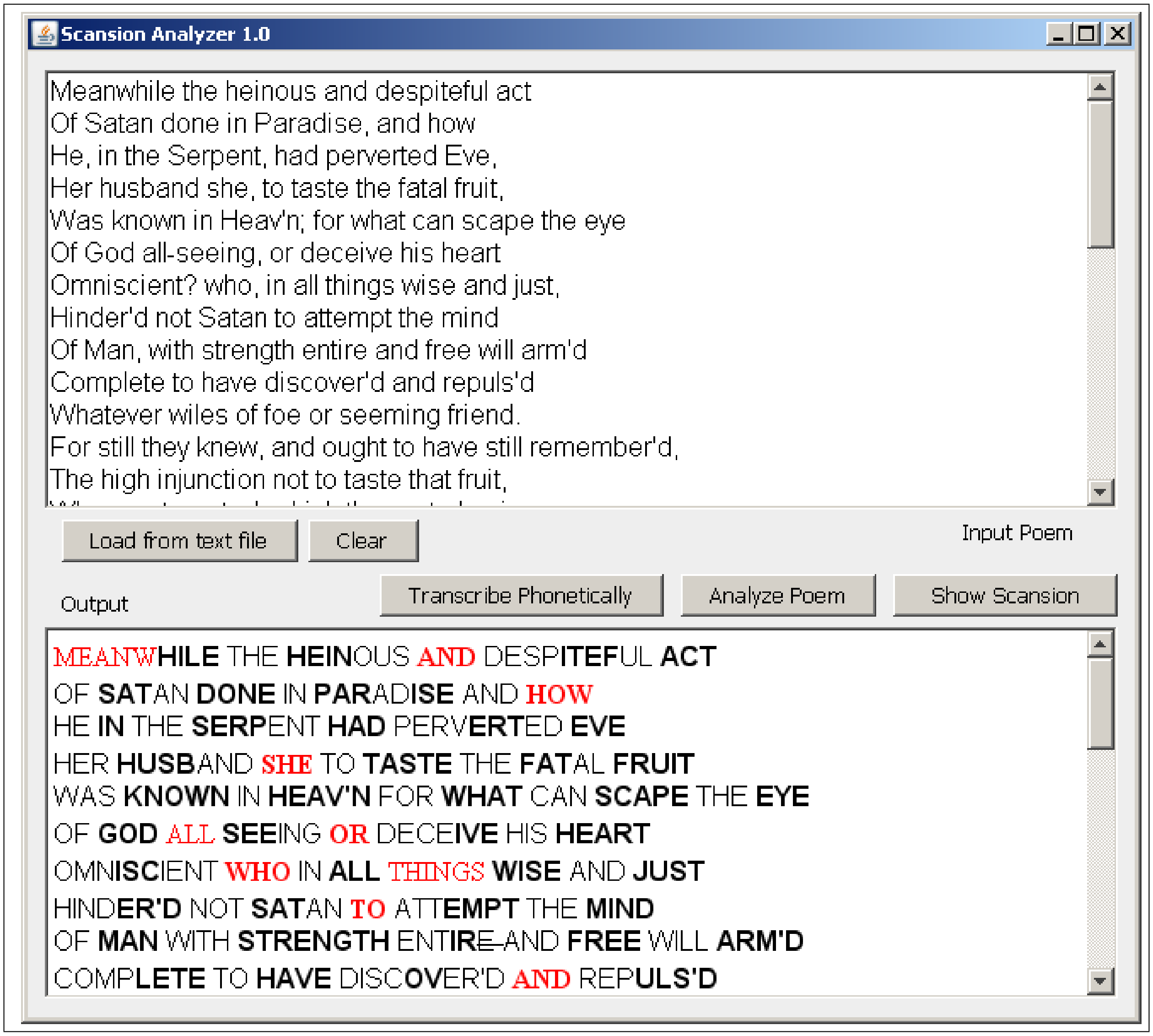

Figure 4.4: A visualization of the detected meter from John Milton's "Paradise Lost: Book X" [75]. Heavily stressed syllables are displayed in bold face font, and syllables on beats not matching their expected stress are coloured red.

Chapter 5

Conclusion

In this thesis, we have quantitatively studied an aspect of text often ignored in Information Retrieval research: the sound of words, realized in their delivery and pronunciation, and modeled computationally by their phonetic transcriptions. This focus on sound allowed us to investigate features such as acoustic similarity, rhyme, and prosodic rhythm. Using sequence analysis methodology characteristic of bioinformatics, in which large corpora of sequences known to be related are used to train probabilistic models of homology, we developed models for these features of lyrical verse integral to its form and structure, and also to its appreciation by listeners or readers.

In Chapter 2, we introduced a probabilistic model of mishearing in sung lyrics. This model was trained using phoneme confusion frequencies calculated from alignments of actual misheard lyrics with their correct counterparts. We collected these lyrics from submission-based misheard lyrics websites, so that the log-odds score for any pair of phonemes a and b indicated how likely it was for a music listener to have heard a when b was actually sung. We discovered that voicing and airflow stoppage were the least salient articulatory features of consonants for music listeners to distinguish as voiced/unvoiced and plosive/fricative consonant pairs with the same place of articulation were most often confused. Vowel height (determined in part by the openness of the singer's mouth) was the most difficult feature of vowels for listeners to distinguish.

Using the model's similarity scores to perform phoneme alignment pattern matching, we were better able to resolve misheard lyric queries than simpler methods such as phoneme edit distance and Syllable Alignment Pattern Search [41]. Tested on 146 misheard lyric queries with correct target lyrics in a collection of 2,345 songs, the probabilistic phoneme model produced a Mean Reciprocal Rank of 0.774, with 108 (74%) of the correct lyrics ranked first in the search results. The model found up to 8% more correct lyrics than the previous best method, phoneme edit distance, which achieved an MRR of 0.709. We analyzed the misheard lyric queries for which the correct lyric was not ranked at all and concluded that many of the pairs were remarkably dissimilar. In others, we observed that listeners sometimes superfluously heard the singer's name, as in "Freddie time!" in a Queen song. For the shortest misheard lyrics, partial exact matches in non-target songs tended to dominate the similarity scores. While longer misheard lyrics were more likely to have their correct targets found across all methods, the probabilistic model was more resistant to this phenomenon and best able to resolve the shortest queries. Finally, using a phoneme trigram index, we were able to reduce the running time per query by more than 50% with a moderate 12% loss in search accuracy, resulting in performance similar to phoneme edit distance in less than half the time.

In Chapter 3, we developed a probabilistic model to identify and score both perfect and imperfect rhymes in rap lyrics. Trained on a corpus of 40 influential "Golden Age" hip hop albums (mostly containing line-final couplet rhymes), the model identified end rhymes with a higher level of accuracy then simpler rules-based methods. We then designed an algorithm using this scoring method to find the nearest and longest rhymes for words in rap lyrics. On an evaluation set of six manually-annotated rap songs of diverse styles, the best performance of the heuristic rhyme detection method achieved sensitivity and specificity just over 60%, meaning around two thirds of annotated pairs were detected as rhyming and two thirds of detected rhymes were annotated as such.

Using the detection algorithm, we were able to calculate statistical features of detected rhymes which corresponded to real world characterizations of rhyme style. Unsurprisingly, we found that popular hip hop songs (from the Billboard Hot Rap charts) contained more rhyme than Hot Rock songs and unrhymed verse, and song lyrics had higher scoring end rhymes than unrhymed verse. We calculated rhyme features for popular albums by 25 important rappers and discovered that modern rappers tended to use more rhyme, longer rhymes, more internal rhymes, and less perfect rhymes, as all of these features varied with time. Many of these features were consistent enough within individual artists' lyrics and varied enough between different artists to allow for rhyme-based artist classification and stylometry. We used the Weka Data Mining Software to build a simple logistic regression classifier trained on statistical rhyme features from the examined albums. In a ten-fold cross-validation, the model achieved classification accuracy over 50% with an F-measure of 0.516, correctly identifying the rapper for 314 out of 603 songs.

We proposed the use of statistical rhyme characterization as a viable indication of an MC's style by considering that the most stylistically diverse rappers were the most resistant to classification. We also identified a particular case ("Notorious Thugs" featuring Bone Thugs-n-Harmony [14]) in which a rapper (The Notorious B.I.G.) deliberately and measurably modified his rhyming style to match another artist. We demonstrated the possibility that rhyme style-based features can be used to identify ghostwritten songs, with a small experiment in which the rhyme feature-based classifier correctly identified the writer of more ghostwritten songs than a naive Bayes bag-of-words model. We examined

Eminem's lyrics over the course of his career and identified trends indicating that his rhymes may have become less technically complex as he became more commercially successful. In examining the relationship between rhyme features and critical and commercial success, we found that both casual music listeners and professional critics preferred rap lyrics with longer words and shorter rhymes, but albums with shorter words and more imperfect end rhymes tended to sell more copies. Finally, we presented a graphical user interface tool with the ability to perform phonetic transcriptions, score and display detected rhymes, calculate statistical rhyme features, and perform MC classification for input rap lyrics.

In Chapter 4, we presented an algorithm to perform automatic scansion and produce a correspondence score to quantify how well a poem aligns with the underlying metrical structure. We used this algorithm to identify metrical poetry in the University of Toronto Library's Representative Poetry Online [83] collection of over 3,000 poems. Restricting the collection to about 750 poems with fixed metrical structures about which we were confident, we characterized words by their likelihood of being stressed. We found that the most common words were less likely to receive heavy stress, and the least often stressed of these tended to be possessive adjectives like "my" or "your." We used these likelihoods to develop a robust method of performing probabilistic scansion that allowed us to more than double the amount of identified metrical poetry to over 1,700 poems.

We compared different poets' adherence to strict metrical forms and found that correspondence scores tended to increase over time as English evolved to more closely resemble the modern pronunciations in our phonetic dictionary. However, we did find significant variations between poets who wrote during the same time period, indicating differences in their use of metrical complexity. Using Whissell's Dictionary of Affect in Language [109], we found that heavily stressed words falling on light beats tended to be more emotionally charged and had better imagery. In comparing poetic rhyme with our model of rhyme from hip hop lyrics, we found more constraints on consonant identity in rhyme, as well as evidence of poets deliberately using eye-rhymes with similarly spelled (but differently pronounced) words. Finally, we presented a graphical user interface tool allowing for phonetic transcription, probabilistic scansion analysis, and the identification and visualization of metrical structure for input poetry.

Prior to our work, analysis of the sound and structure of lyrical texts was a slow and mostly descriptive process, with researchers making painstaking case studies and producing subjective, qualitative results about the form, effect, composition, or similarity of different works. Automated and quantitative methods involved simple or generic models of the features being studied. Our probabilistic methods allow for quick, quantitative, and domain-specific characterizations of large collections of data. Our software tools' simple user-interfaces allow anyone to easily perform efficient and accurate analyses of lyrical texts. The Rap Analyzer could be used by a hip hop listener to better appreciate the complexity of an MC's rhyming scheme, by a record executive to quickly characterize a potential artist's output, or even by an aspiring rapper to break down and improve upon her own lyrics. The Scansion Analyzer could be used by English instructors as a teaching tool to discover and demonstrate meter, or by poetry readers to quantify and compare poets' metrical complexity.

Although each chapter in this thesis focuses primarily on a different aspect of lyrical verse (acoustic similarity in misheard song lyrics, rhymes in hip hop, and rhythmic meter in conventional poetry), an ideal application would allow for a unified analysis including all of these features. Similar to the statistical features we used to characterize rhyming style in hip hop, we could calculate relevant statistics on patterns in metrical form and complexity, modified prosody, emotional and figurative language, and the usage of various rhyme types and schemes to better quantitatively characterize poetic style

While it would not be particularly difficult to more thoroughly investigate rhyme in poetry, a more interesting challenge would be to statistically characterize rhythm and metric style in rap. Though rarely having fixed syllable counts per line, rap songs often have regular (though complex) stress patterns, with a constrained but variable number of lightly stressed syllables per metric "foot." Identifying these patterns would aid in the detection of rhymes, especially when the rapper modifies the pronunciation of words to fit the rhythm.

Furthermore, while rhyme is the integral feature of hip hop music, another crucial aspect of an MC's unique style is his "flow," referring to the rhythm, dynamics, and intonation of his vocal delivery and its relation to the accompanying beat [17]. For example, Eminem's debut *Infinite* [33], was classified as Eminem using the rhyme feature logistic regression model described in Section 3.7, but his flow on the album was considered to be derivative of rappers Nas and AZ. He has been quoted as saying "obviously, I was young and influenced by other artists, and I got a lot of feedback saying that I [sounded a lot like] Nas and AZ" [74]. Upon listening, one can identify similarities in the syncopated rhythm of rapped syllables in, for example, "Life's a Bitch" (featuring AZ) from Nas' *Illmatic* and the title track from *Infinite*, and we suspect that these similarities should be quantifiable.

Both of these applications suggest the requirement of integrating rhyme and prosodic features extracted from text lyrics with rhythm and pronunciation information drawn from audio recordings using a speech recognition engine. Lewis and Assogba [66] presented an artistic performance in which rapped words were recognized by Nuance's Dragon Naturally Speaking commercial software to form a text visualization in near real time. They reported a recognition accuracy rate of 75% after training, but our own experiments (which we do not discuss in this thesis) with a capella versions of rap songs proved far less successful. A more fruitful method would likely involve determining the rhythmic position of syllables without identifying words, essentially treating the voice as a percussive instrument and working with vocal beats in a style similar to Dixon et al's [25] work in discriminating rhythmic patterns in ballroom dance music. This sort of approach could allow for the quantitative characterization of different MCs' flows and allow for a unified analysis of rhyme and rhythm.

Bibliography

[1] 2Pac. Me Against the World. Jive Records, 1995.

[2] 2Pac. All Eyez on Me. Death Row Records, 1996.

[3] H. Adaso. 50 Greatest MCs of Our Time (1987-2007). `http://rap.about.com/od/toppicks/ss/Top50Emcees.htm`, 2007.

[4] H.S. Alim. On Some Serious Next Millennium Rap Ishhh: Pharoahe Monch, Hip Hop poetics, and the Internal Rhymes of Internal Affairs. *Journal of English Linguistics*, 31(1):60, 2003.

[5] D.P. Alridge and J.B. Stewart. Introduction: Hip Hop in history: Past, present, and future. *The Journal of African American History*, 90(3):190–195, 2005.

[6] S.F. Altschul, W. Gish, W. Miller, E.W. Myers, and D.J. Lipman. Basic local alignment search tool. *Journal of Molecular Biology*, 215(3):403–410, 1990.

[7] American Society of Composers, Authors, and Publishers. Works written by Lewis, Donnie Ray. `http://ascap.com/ace/search.cfm?mode=search`, 2010.

[8] Matthew Arnold. *Empedocles on Etna, and Other Poems.* B. Fellowes, 1852.

[9] International Phonetic Association. Handbook of the International Phonetic Association: a guide to the use of the International Phonetic Alphabet. Cambridge: Cambridge University Press, 1999.

[10] Eric B. and Rakim. Paid in Full. 4th and B'way Records, 1987.

[11] Eric B. and Rakim. Follow the Leader. Uni Records, 1988.

[12] Eric B. and Rakim. Let the Rhythm Hit 'Em. MCA Records, 1990.

[13] The Notorious B.I.G. Ready to Die. Bad Boy Records, 1994.

[14] The Notorious B.I.G. Life After Death. Bad Boy Records, 1997.

[15] Beastie Boys. Licensed To Ill. Def Jam Recordings, 1986.

[16] Beastie Boys. Paul's Boutique. Def Jam Recordings, 1989.

[17] A. Bradley. *Book of Rhymes: The Poetics of Hip Hop*. Nation Books, 2009.

[18] Adam Bradley. personal communication, 2010.

[19] Elizabeth Barrett Browning. *Poems*. Edward Moxon, 1844.

[20] PY Chou and GD Fasman. Prediction of the secondary structure of proteins from their amino acid sequence. *Advances in enzymology and related areas of molecular biology*, 47:45, 1978.

[21] Ice Cube. AmeriKKKa's Most Wanted. Priority Records, 1990.

[22] Ice Cube. Death Certificate. Priority Records, 1990.

[23] John Denham. *Coopers Hill. A Poeme. The Second Edition with Additions.* Humphrey Moseley, 1950.

[24] R.J. Dilligan and K. Lynn. Computers and the History of Prosody. *College English*, 34(8):1103–1123, 1973.

[25] S. Dixon, F. Gouyon, and G. Widmer. Towards characterisation of music via rhythmic patterns. In *Proceedings of the International Conference on Music Information Retrieval*, 2004.

[26] DMX. It's Dark and Hell is Hot. Ruff Ryders Entertainment, 1998.

[27] DMX. ...And Then There Was X. Ruff Ryders Entertainment, 1999.

[28] H.S. Donow. Prosody and the computer: a text processor for stylistic analysis. In *Proceedings of the May 5-7, 1970, Spring Joint Computer Conference*, pages 287–295, 1970.

[29] Dr. Dre. 2001. Aftermath Entertainment, 1999.

[30] R. Durbin, S.R. Eddy, A. Krogh, and G. Mitchison. *Biological sequence analysis: Probabilistic models of proteins and nucleic acids*. Cambridge Univ Press, 1998.

[31] Eazy-E. Boyz-N-The-Hood. Ruthless Records, 1987.

[32] H. Elovitz, R. Johnson, A. McHugh, and J. Shore. Letter-to-sound rules for automatic translation of English text to phonetics. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 24(6):446–459, 1976.

[33] Eminem. Infinite, 1999.

[34] Eminem. The Slim Shady LP. Aftermath Entertainment, 1999.

[35] Eminem. The Marshall Mathers LP. Aftermath Entertainment, 2000.

[36] Public Enemy. It Takes a Nation of Millions to Hold Us Back. Def Jam Recordings, 1988.

[37] Public Enemy. Fear of a Black Planet. Def Jam Recordings, 1990.

[38] Fabolous. Ghetto Fabolous. Elektra Records, 2001.

[39] Fabolous. Street Dreams. Desert Storm Records, 2003.

[40] H. Fujihara, M. Goto, and J. Ogata. Hyperlinking lyrics: A method for creating hyperlinks between phrases in song lyrics. In *Proceedings of the International Conference on Music Information Retrieval*, pages 281–286, 2008.

[41] R. Gong and T.K.Y. Chan. Syllable alignment: A novel model for phonetic string search. *IEICE Transactions on Information and Systems*, 89(1):332–339, 2006.

[42] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I.H. Witten. The WEKA data mining software: An update. *ACM SIGKDD Explorations Newsletter*, 11(1):10–18, 2009.

[43] M. Halle and S.J. Keyser. *English stress: its form, its growth, and its role in verse.* Harper & Row, 1971.

[44] MC Hammer. Please Hammer, Don't Hurt 'Em. Capitol Records, 1990.

[45] MC Hammer. Too Legit to Quit. Capitol Records, 1991.

[46] S. Henikoff and JG Henikoff. Amino acid substitution matrices from protein blocks. *Proceedings of the National Academy of Sciences of the United States of America*, 89(22):10915, 1992.

[47] H. Hirjee and D.G. Brown. Automatic Detection of Internal and Imperfect Rhymes in Rap Lyrics. In *Proceedings of the 10th International Society for Music Information Retrieval Conference*, 2009.

[48] H. Hirjee and D.G. Brown. Rhyme Analyzer: An Analysis Tool for Rap Lyrics. In *Proceedings of the 11th International Society for Music Information Retrieval Conference*, 2010.

[49] H. Hirjee and D.G. Brown. Solving Misheard Lyric Search Queries Using a Probabilistic Model of Speech Sounds. In *Proceedings of the 11th International Society for Music Information Retrieval Conference*, 2010.

[50] H. Hirjee and D.G. Brown. Using Automated Rhyme Detection to Characterize Rhyming Style in Rap Music. *Empirical Musicology Review*, 2010. Under review.

[51] A Holtman. *A Generative Theory of Rhyme: An Optimality Approach*. PhD thesis, Utrecht Institute of Linguistics, 1996.

[52] G. Hripcsak and A.S. Rothschild. Agreement, the f-measure, and reliability in information retrieval. *Journal of the American Medical Informatics Association*, 12(3):296, 2005.

[53] LL Cool J. Radio. Def Jam Recordings, 1985.

[54] LL Cool J. Bigger and Deffer. Def Jam Recordings, 1987.

[55] Jay-Z. Reasonable Doubt. Roc-A-Fella Records, 1996.

[56] Jay-Z. The Blueprint. Roc-A-Fella Records, 2001.

[57] Big Daddy Kane. Long Live the Kane. Cold Chillin' Records, 1988.

[58] Big Daddy Kane. It's a Big Daddy Thing. Cold Chillin' Records, 1990.

[59] S. Kawahara. Half rhymes in Japanese rap lyrics and knowledge of similarity. *Journal of East Asian Linguistics*, 16(2):113–144, 2007.

[60] F. Kleedorfer, P. Knees, and T. Pohle. Oh oh oh whoah! towards automatic topic detection in song lyrics. In *Proceedings of the International Conference on Music Information Retrieval*, pages 287–292, 2008.

[61] G. Kondrak. A new algorithm for the alignment of phonetic sequences. In *Proceedings of the 1st North American chapter of the Association for Computational Linguistics conference*, pages 288–295, 2000.

[62] A. Krims. *Rap music and the poetics of identity*. Cambridge Univ Pr, 2000.

[63] Henry Lawson. *When I was King and Other Verses*. Angus and Robertson, 1906.

[64] K. Lenzo. The CMU Pronouncing Dictionary. `http://www.speech.cs.cmu.edu/cgi-bin/cmudict`, 2008.

[65] V.I. Levenshtein. Binary codes capable of correcting spurious insertions and deletions of ones. *Problems of Information Transmission*, 1(1):8–17, 1965.

[66] J. Lewis and Y. Assogba. Taking sides: dynamic text and hip-hop performance. In *Proceedings of the 14th annual ACM international conference on Multimedia*, page 747. ACM, 2006.

[67] B.T. Logan and A. Salomon. Music similarity function based on signal analysis, October 31, 2001. US Patent App. 10/004,157.

[68] H.M. Logan. Computer and metrical scansion. *ALLC J.*, 3(1):9–14, 1982.

[69] XXL Magazine. The Making of Life After Death: Many Men. `http://www.xxlmag.com/online/?p=405`, 2003.

[70] XXL Magazine. Skillz: I'm the Dopest. `http://www.xxlmag.com/online/?p=12639`, 2007.

[71] M. Marolt. A mid-level melody-based representation for calculating audio similarity. In *Proceedings of the International Conference on Music Information Retrieval*, pages 280–285. Citeseer, 2006.

[72] R. Mayer, R. Neumayer, and A. Rauber. Rhyme and style features for musical genre classification by song lyrics. In *Proceedings of the International Conference on Music Information Retrieval*, pages 337–342, 2008.

[73] J.M. McQueen, D. Norris, and A. Cutler. Competition in spoken word recognition: Spotting words in other words. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 20(3):621–638, 1994.

[74] S. Michaels. Eminem's rare debut available for free download. *The Guardian*, Thursday, May 14, 2009.

[75] J. Milton. *Paradise Lost*. Samuel Simmons, 1667.

[76] T. Mitchell. *Global noise: Rap and hip-hop outside the USA*. Wesleyan Univ Pr, 2002.

[77] Pharoahe Monch. Internal Affairs. Rawkus Records, 1999.

[78] MTV. The Greatest MCs of All Time. `http://www.mtv.com/bands/h/hip_hop_week/2006/emcees/`, 2006.

[79] Bone Thugs n Harmony. E. 1999 Eternal. Ruthless Records, 1995.

[80] N.W.A. Straight Outta Compton. Ruthless Records, 1988.

[81] The Recording Industry Association of America. Top Selling Artists. `http://www.riaa.com/goldandplatinumdata.php?table=tblTopArt`, 2010.

[82] Scarface of the Geto Boys. Mr. Scarface is Back. Rap-a-Lot Records, 1991.

[83] Representative Poetry Online. Online Editor's Introduction (Version 3.0). `http://rpo.library.utoronto.ca/display_rpo/intro.cfm`, 2010.

[84] Outkast. Southernplayalisticadillacmuzik. LaFace Records, 1994.

[85] Outkast. ATLiens. LaFace Records, 1996.

[86] Outkast. Aquemini. LaFace Records, 1998.

[87] E. Pampalk, A. Flexer, and G. Widmer. Improvements of audio-based music similarity and genre classification. In *Proceedings of the International Conference on Music Information Retrieval*, pages 628–633, 2005.

[88] M.R. Plamondon. Virtual Verse Analysis: Analysing Patterns in Poetry. *Literary and Linguistic Computing*, 21(Supplement 1):127, 2006.

[89] Boogie Down Productions. Criminal Minded. B-Boy Records, 1987.

[90] Boogie Down Productions. By All Means Necessary. Jive Records, 1988.

[91] Kool G Rap and DJ Polo. Road to the Riches. Cold Chillin' Records, 1989.

[92] Kool G Rap and DJ Polo. Wanted: Dead or Alive. Cold Chillin' Records, 1990.

[93] Slick Rick. The Ruler's Back. Def Jam Recordings, 1991.

[94] N. Ring and A. Uitdenbogerd. Finding Lucy in Disguise: The Misheard Lyric Matching Problem. *Lecture Notes in Computer Science*, 5839:157–167, 2009.

[95] E.S. Ristad and P.N. Yianilos. Learning string-edit distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(5):522–532, 1998.

[96] R. Ro. *Raising hell: the reign, ruin, and redemption of Run-D.M.C. and Jam Master Jay*. Amistad, 2005.

[97] Run-D.M.C. Run-D.M.C. Profile Records, 1984.

[98] Run-D.M.C. Raising Hell. Profile Records, 1986.

[99] Run-D.M.C. Tougher Than Leather. Profile Records, 1988.

[100] N. Saitou and M. Nei. The neighbor-joining method: a new method for reconstructing phylogenetic trees. *Molecular Biology and Evolution*, 4(4):406, 1987.

[101] William Shakespeare. *SHAKE-SPEARES SONNETS*. Thomas Thorpe, 1609.

[102] Will Smith. Big Willie Style. Columbia Records, 1997.

[103] Will Smith. Lost and Found. Interscope Records, 2005.

[104] G.C. Terstappen and A. Reggiani. In silico research in drug discovery. *Trends in pharmacological sciences*, 22(1):23–26, 2001.

[105] Timbaland and Magoo. Indecent Proposal. Blackground Records, 2001.

[106] Lil' Wayne. Tha Carter II. Cash Money Records, 2005.

[107] Lil' Wayne. Tha Carter III. Cash Money Records, 2008.

[108] B. Wei, U. Rochester, C. Zhang, and M. Ogihara. Keyword generation for lyrics. In *Proceedings of the International Conference on Music Information Retrieval*, pages 121–122, 2007.

[109] C.M. Whissell. The dictionary of affect in language. *Emotion: Theory, Research and Experience*, 4:113–131, 1989.

[110] Cynthia Whissell. Whissell's Dictionary of Affect in Language: Technical Manual and User's Guide. `http://www.hdcus.com/manuals/wdalman.pdf`, 2010.

[111] John Wilmot. *Poems on Several Occasions*. Antwerp, 1680.

[112] S. Wright. The death of Lady Mondegreen. *Harpers Magazine*, 209:48–51, 1954.

[113] X. Xu, M. Naito, T. Kato, and H. Kawai. Robust and Fast Lyric Search Based on Phonetic Confusion Matrix. In *Proceedings of the International Conference on Music Information Retrieval*, pages 417–422, 2009.

[114] I. Youngs. 50 Cent: The $1m rapper. *BBC News - Entertainment*, Monday, December 23, 2002. `http://news.bbc.co.uk/2/hi/entertainment/2591521.stm`.

Appendix: Full List of Rhyme Features

Artist	Album	Year	Syllables per Line	Syllables per Word	Syllable Variation	Novel Word Proportion
Run-D.M.C.	Run-D.M.C.	1984	10.40	1.21	1.82	0.91
Run-D.M.C.	Raising Hell	1986	10.88	1.21	2.05	0.91
Run-D.M.C.	Tougher Than Leather	1988	10.17	1.21	2.21	0.89
LL Cool J	Radio	1985	11.49	1.29	2.03	0.92
LL Cool J	Bigger and Deffer	1987	11.71	1.23	1.89	0.92
Beastie Boys	Licensed To Ill	1986	10.81	1.21	1.98	0.88
Beastie Boys	Paul's Boutique	1989	11.30	1.27	2.38	0.90
Rakim	Paid in Full	1987	11.18	1.24	1.93	0.90
Rakim	Follow the Leader	1988	10.83	1.32	2.29	0.92
Rakim	Let the Rhythm Hit 'Em	1990	10.62	1.27	1.94	0.92
KRS-One	Criminal Minded	1987	11.77	1.30	2.14	0.92
KRS-One	By All Means Necessary	1988	11.20	1.30	2.72	0.91
Chuck D	It Takes a Nation of Millions to Hold Us Back	1988	10.35	1.26	2.42	0.90
Chuck D	Fear of a Black Planet	1990	9.78	1.25	2.43	0.92
Big Daddy Kane	Long Live the Kane	1988	10.76	1.29	1.90	0.92
Big Daddy Kane	It's a Big Daddy Thing	1989	10.46	1.25	2.03	0.93
Slick Rick	The Great Adventures of Slick Rick	1988	11.40	1.20	2.11	0.91
Slick Rick	The Ruler's Back	1991	12.62	1.26	2.65	0.91
Kool G Rap	Road to the Riches	1989	10.94	1.32	2.04	0.93
Kool G Rap	Wanted: Dead or Alive	1990	10.96	1.29	2.47	0.92
Ice Cube	AmeriKKKa's Most Wanted	1990	10.04	1.20	2.21	0.91
Ice Cube	Death Certificate	1991	9.65	1.24	2.58	0.93
MC Hammer	Please Hammer Don't Hurt 'Em	1990	8.93	1.21	3.13	0.92
MC Hammer	Too Legit to Quit	1991	10.94	1.21	2.57	0.91
Scarface	Mr. Scarface is Back	1991	10.37	1.22	2.90	0.90
Scarface	The Diary	1994	11.81	1.22	2.87	0.88
Redman	Whut? Thee Album	1992	11.69	1.20	2.21	0.91
Redman	Muddy Waters	1996	11.33	1.27	2.02	0.93
Nas	Illmatic	1994	12.45	1.27	2.42	0.92
Nas	It Was Written	1996	11.84	1.30	2.62	0.92
Nas	I Am...	1999	12.37	1.26	2.05	0.91
Andre 3000	Southernplayalisti-cadillacmuzik	1994	14.58	1.37	3.25	0.92
Andre 3000	ATLiens	1996	14.68	1.35	3.31	0.91
Andre 3000	Aquemini	1998	14.76	1.48	5.02	0.91
The Notorious B.I.G.	Ready to Die	1994	11.08	1.26	2.31	0.92
The Notorious B.I.G.	Life After Death	1997	10.62	1.25	2.52	0.92
2Pac	Me Against the World	1995	12.35	1.30	2.50	0.91
2Pac	All Eyez on Me	1996	12.46	1.29	2.61	0.92
Bone Thugs-n-Harmony	E. 1999 Eternal	1995	17.23	1.24	3.22	0.89
Jay-Z	Reasonable Doubt	1996	12.08	1.28	2.39	0.92
Jay-Z	In My Lifetime, Vol. 1	1997	12.13	1.22	2.23	0.90
Jay-Z	Vol. 2: Life and Times of Shawn Carter	1998	11.64	1.22	2.32	0.89
DMX	It's Dark and Hell is Hot	1998	12.30	1.18	2.44	0.88
DMX	...And Then There Was X	1999	12.32	1.19	2.24	0.87
Eminem	The Slim Shady LP	1999	12.41	1.24	2.50	0.88
Eminem	The Marshall Mathers LP	2000	12.39	1.23	2.55	0.88
Nelly	Country Grammar	2000	11.58	1.23	2.30	0.89
Nelly	Nellyville	2002	11.81	1.23	2.55	0.88
Fabolous	Ghetto Fabolous	2001	11.45	1.21	2.24	0.89
Fabolous	Street Dreams	2003	11.47	1.24	2.42	0.87
50 Cent	Get Rich or Die Tryin'	2003	11.73	1.20	2.48	0.88
50 Cent	The Massacre	2005	11.57	1.20	2.71	0.90
Lil' Wayne	Tha Carter II	2005	11.45	1.27	2.66	0.88
Lil' Wayne	Tha Carter III	2008	10.71	1.24	2.92	0.87

Artist	Year	Rhymes per Line	Rhymes per Syllable	Rhyme Density	End Pairs per Line	End Pairs Grown	End Pairs Shrunk	End Pairs Even
Run-D.M.C.	1984	1.48	0.14	0.20	0.54	0.30	0.17	0.53
Run-D.M.C.	1986	1.96	0.18	0.26	0.47	0.26	0.18	0.56
Run-D.M.C.	1988	2.51	0.25	0.42	0.43	0.21	0.18	0.62
LL Cool J	1985	1.68	0.15	0.20	0.46	0.27	0.18	0.55
LL Cool J	1987	1.87	0.16	0.21	0.47	0.37	0.16	0.48
Beastie Boys	1986	1.63	0.15	0.22	0.50	0.24	0.20	0.56
Beastie Boys	1989	1.67	0.15	0.22	0.47	0.29	0.25	0.46
Rakim	1987	1.64	0.15	0.23	0.51	0.23	0.29	0.48
Rakim	1988	1.85	0.17	0.27	0.46	0.25	0.32	0.43
Rakim	1990	1.76	0.17	0.25	0.44	0.21	0.25	0.54
KRS-One	1987	1.74	0.15	0.22	0.42	0.18	0.19	0.63
KRS-One	1988	1.90	0.17	0.26	0.43	0.26	0.28	0.46
Chuck D	1988	1.75	0.17	0.25	0.33	0.28	0.23	0.49
Chuck D	1990	1.66	0.17	0.24	0.31	0.27	0.30	0.43
Big Daddy Kane	1988	1.71	0.16	0.24	0.44	0.29	0.19	0.52
Big Daddy Kane	1989	1.82	0.17	0.26	0.42	0.31	0.23	0.46
Slick Rick	1988	1.86	0.16	0.23	0.48	0.36	0.16	0.48
Slick Rick	1991	2.25	0.18	0.29	0.34	0.28	0.12	0.59
Kool G Rap	1989	2.13	0.19	0.29	0.34	0.29	0.21	0.50
Kool G Rap	1990	2.08	0.19	0.29	0.35	0.34	0.30	0.36
Ice Cube	1990	1.35	0.13	0.19	0.44	0.31	0.29	0.40
Ice Cube	1991	1.30	0.13	0.19	0.48	0.49	0.22	0.29
MC Hammer	1990	1.30	0.15	0.20	0.46	0.51	0.13	0.36
MC Hammer	1991	1.51	0.14	0.19	0.45	0.37	0.17	0.46
Scarface	1991	1.37	0.13	0.20	0.50	0.55	0.22	0.24
Scarface	1994	1.69	0.14	0.24	0.47	0.45	0.19	0.36
Redman	1992	2.11	0.18	0.26	0.34	0.26	0.21	0.53
Redman	1996	1.89	0.17	0.24	0.37	0.32	0.19	0.49
Nas	1994	2.32	0.19	0.29	0.33	0.31	0.18	0.50
Nas	1996	2.17	0.18	0.29	0.38	0.30	0.34	0.36
Nas	1999	2.35	0.19	0.29	0.36	0.27	0.24	0.49
Andre 3000	1994	3.28	0.22	0.34	0.32	0.41	0.21	0.38
Andre 3000	1996	2.78	0.19	0.29	0.31	0.30	0.19	0.51
Andre 3000	1998	3.56	0.24	0.40	0.29	0.46	0.24	0.30
The Notorious B.I.G.	1994	1.94	0.17	0.26	0.31	0.38	0.25	0.36
The Notorious B.I.G.	1997	2.00	0.19	0.30	0.35	0.26	0.29	0.44
2Pac	1995	1.96	0.16	0.24	0.32	0.25	0.24	0.51
2Pac	1996	1.85	0.15	0.22	0.38	0.26	0.27	0.46
Bone Thugs-n-Harmony	1995	3.58	0.21	0.33	0.21	0.27	0.27	0.47
Jay-Z	1996	2.15	0.18	0.29	0.28	0.30	0.18	0.52
Jay-Z	1997	2.11	0.17	0.27	0.42	0.22	0.19	0.59
Jay-Z	1998	2.18	0.19	0.32	0.35	0.24	0.27	0.49
DMX	1998	2.06	0.17	0.30	0.33	0.28	0.16	0.55
DMX	1999	2.11	0.17	0.29	0.33	0.21	0.23	0.57
Eminem	1999	2.17	0.17	0.31	0.44	0.29	0.22	0.49
Eminem	2000	2.40	0.19	0.34	0.43	0.24	0.18	0.58
Nelly	2000	2.09	0.18	0.28	0.43	0.23	0.14	0.63
Nelly	2002	1.95	0.17	0.27	0.40	0.24	0.17	0.59
Fabolous	2001	1.86	0.16	0.35	0.34	0.25	0.32	0.43
Fabolous	2003	1.92	0.17	0.35	0.32	0.25	0.24	0.51
50 Cent	2003	1.82	0.16	0.25	0.44	0.21	0.18	0.61
50 Cent	2005	1.75	0.15	0.22	0.45	0.19	0.20	0.61
Lil' Wayne	2005	2.21	0.19	0.33	0.40	0.32	0.26	0.42
Lil' Wayne	2008	1.92	0.18	0.32	0.44	0.33	0.30	0.38

Artist	Year	Average End Score	Average End Syl Score	Singles per Rhyme	Doubles per Rhyme	Triples per Rhyme	Quads per Rhyme	Longs per Rhyme
Run-D.M.C.	1984	5.51	4.59	74%	20%	4%	1%	2%
Run-D.M.C.	1986	5.58	4.49	67%	26%	4%	2%	1%
Run-D.M.C.	1988	6.18	4.27	59%	24%	10%	4%	3%
LL Cool J	1985	5.28	4.28	75%	18%	4%	1%	1%
LL Cool J	1987	5.26	4.58	74%	21%	4%	1%	0%
Beastie Boys	1986	5.99	4.68	70%	21%	4%	4%	1%
Beastie Boys	1989	6.57	4.52	64%	27%	6%	2%	1%
Rakim	1987	6.10	4.07	63%	24%	9%	2%	1%
Rakim	1988	6.10	3.96	59%	28%	9%	3%	1%
Rakim	1990	6.00	3.85	62%	28%	7%	2%	1%
KRS-One	1987	5.88	4.10	65%	28%	5%	2%	1%
KRS-One	1988	5.46	3.87	65%	25%	6%	3%	2%
Chuck D	1988	5.46	4.00	67%	24%	6%	2%	0%
Chuck D	1990	5.40	3.89	67%	26%	6%	2%	0%
Big Daddy Kane	1988	6.33	4.13	63%	27%	7%	2%	1%
Big Daddy Kane	1989	6.04	4.01	62%	28%	8%	2%	1%
Slick Rick	1988	5.83	4.40	72%	21%	4%	2%	1%
Slick Rick	1991	6.40	3.80	58%	29%	8%	3%	2%
Kool G Rap	1989	6.49	4.25	62%	29%	6%	2%	0%
Kool G Rap	1990	6.33	4.08	58%	33%	7%	2%	1%
Ice Cube	1990	6.04	4.16	67%	25%	5%	1%	1%
Ice Cube	1991	6.08	4.01	67%	26%	5%	2%	1%
MC Hammer	1990	4.74	4.14	74%	18%	5%	2%	1%
MC Hammer	1991	5.13	4.16	70%	24%	4%	2%	0%
Scarface	1991	5.69	3.92	63%	30%	5%	2%	1%
Scarface	1994	6.35	3.87	59%	28%	7%	3%	3%
Redman	1992	5.01	3.41	66%	27%	4%	2%	1%
Redman	1996	5.09	3.51	66%	27%	5%	1%	0%
Nas	1994	5.91	3.77	60%	30%	7%	3%	1%
Nas	1996	5.36	3.44	57%	33%	6%	3%	1%
Nas	1999	5.03	3.46	63%	27%	5%	4%	1%
Andre 3000	1994	5.05	3.68	64%	27%	5%	2%	1%
Andre 3000	1996	5.44	3.84	65%	26%	6%	3%	1%
Andre 3000	1998	4.93	3.41	64%	22%	7%	5%	3%
The Notorious B.I.G.	1994	5.80	3.90	62%	29%	6%	2%	1%
The Notorious B.I.G.	1997	5.34	3.65	58%	30%	8%	3%	1%
2Pac	1995	5.08	3.58	61%	30%	7%	2%	0%
2Pac	1996	5.69	3.74	60%	31%	6%	2%	0%
Bone Thugs-n-Harmony	1995	4.59	3.47	58%	30%	7%	3%	1%
Jay-Z	1996	5.09	3.36	56%	31%	9%	3%	1%
Jay-Z	1997	5.47	3.63	62%	24%	9%	4%	1%
Jay-Z	1998	5.51	3.64	56%	28%	8%	6%	2%
DMX	1998	6.34	3.83	55%	27%	10%	6%	3%
DMX	1999	6.05	3.88	57%	28%	9%	4%	2%
Eminem	1999	6.47	3.78	54%	27%	12%	5%	2%
Eminem	2000	5.40	3.20	54%	28%	10%	6%	3%
Nelly	2000	4.95	3.74	62%	28%	8%	2%	1%
Nelly	2002	4.98	3.70	61%	27%	7%	3%	2%
Fabolous	2001	6.33	3.41	49%	19%	14%	12%	7%
Fabolous	2003	6.10	3.41	48%	23%	12%	9%	8%
50 Cent	2003	5.96	3.85	62%	26%	7%	3%	2%
50 Cent	2005	5.27	3.87	66%	25%	5%	2%	1%
Lil' Wayne	2005	5.30	3.33	55%	29%	9%	4%	3%
Lil' Wayne	2008	5.43	3.59	53%	30%	9%	5%	3%

Artist	Year	Perfect Rhymes	Line Internals per Line	Links per Line	Bridges per Line	Compounds per Line	Chaining per Line
Run-D.M.C.	1984	27%	0.47	0.16	0.32	0.06	0.21
Run-D.M.C.	1986	22%	0.88	0.19	0.42	0.12	0.23
Run-D.M.C.	1988	17%	1.30	0.27	0.52	0.11	0.41
LL Cool J	1985	16%	0.61	0.22	0.38	0.10	0.18
LL Cool J	1987	18%	0.73	0.19	0.48	0.11	0.25
Beastie Boys	1986	25%	0.55	0.18	0.40	0.07	0.19
Beastie Boys	1989	21%	0.61	0.18	0.41	0.09	0.15
Rakim	1987	17%	0.61	0.13	0.39	0.08	0.13
Rakim	1988	13%	0.73	0.22	0.44	0.09	0.19
Rakim	1990	11%	0.68	0.24	0.40	0.08	0.12
KRS-One	1987	16%	0.62	0.27	0.43	0.10	0.12
KRS-One	1988	17%	0.67	0.27	0.53	0.11	0.22
Chuck D	1988	15%	0.79	0.25	0.38	0.10	0.20
Chuck D	1990	13%	0.79	0.21	0.36	0.11	0.27
Big Daddy Kane	1988	16%	0.71	0.18	0.38	0.07	0.19
Big Daddy Kane	1989	14%	0.80	0.25	0.35	0.10	0.26
Slick Rick	1988	16%	0.63	0.24	0.51	0.11	0.22
Slick Rick	1991	10%	0.89	0.32	0.71	0.17	0.18
Kool G Rap	1989	13%	1.04	0.37	0.38	0.15	0.25
Kool G Rap	1990	13%	1.01	0.32	0.40	0.14	0.22
Ice Cube	1990	18%	0.40	0.19	0.32	0.04	0.13
Ice Cube	1991	15%	0.34	0.18	0.30	0.05	0.12
MC Hammer	1990	23%	0.41	0.12	0.32	0.09	0.17
MC Hammer	1991	18%	0.56	0.21	0.29	0.05	0.20
Scarface	1991	15%	0.33	0.16	0.38	0.05	0.09
Scarface	1994	12%	0.47	0.22	0.53	0.07	0.14
Redman	1992	9%	1.02	0.27	0.48	0.17	0.28
Redman	1996	9%	0.74	0.30	0.48	0.11	0.18
Nas	1994	7%	1.02	0.35	0.63	0.25	0.18
Nas	1996	7%	0.90	0.34	0.56	0.16	0.17
Nas	1999	8%	0.90	0.38	0.71	0.20	0.15
Andre 3000	1994	16%	1.49	0.39	1.07	0.57	0.43
Andre 3000	1996	14%	1.23	0.27	0.98	0.41	0.29
Andre 3000	1998	18%	1.71	0.35	1.22	0.73	0.51
The Notorious B.I.G.	1994	10%	0.83	0.39	0.41	0.13	0.18
The Notorious B.I.G.	1997	9%	0.82	0.37	0.46	0.12	0.27
2Pac	1995	10%	0.69	0.33	0.62	0.14	0.12
2Pac	1996	9%	0.63	0.27	0.57	0.12	0.12
Bone Thugs-n-Harmony	1995	6%	1.85	0.37	1.14	0.65	0.76
Jay-Z	1996	8%	0.83	0.31	0.73	0.16	0.17
Jay-Z	1997	8%	0.74	0.31	0.65	0.12	0.22
Jay-Z	1998	10%	0.80	0.31	0.73	0.13	0.23
DMX	1998	8%	0.70	0.30	0.72	0.14	0.20
DMX	1999	11%	0.74	0.32	0.72	0.11	0.14
Eminem	1999	7%	0.77	0.30	0.67	0.15	0.21
Eminem	2000	8%	0.85	0.36	0.75	0.16	0.18
Nelly	2000	13%	0.74	0.34	0.59	0.14	0.18
Nelly	2002	12%	0.72	0.27	0.55	0.12	0.21
Fabolous	2001	5%	0.50	0.22	0.81	0.06	0.13
Fabolous	2003	7%	0.55	0.18	0.86	0.09	0.16
50 Cent	2003	11%	0.57	0.21	0.61	0.10	0.16
50 Cent	2005	13%	0.56	0.22	0.52	0.10	0.12
Lil' Wayne	2005	12%	0.83	0.34	0.63	0.12	0.29
Lil' Wayne	2008	12%	0.68	0.29	0.52	0.11	0.27