SOLVING MISHEARD LYRIC SEARCH QUERIES USING A PROBABILISTIC MODEL OF SPEECH SOUNDS

Hussein Hirjee **Daniel G. Brown**
University of Waterloo
Cheriton School of Computer Science
{hahirjee, browndg}@uwaterloo.ca

ABSTRACT

Music listeners often mishear the lyrics to unfamiliar songs heard from public sources, such as the radio. Since standard text search engines will find few relevant results when they are entered as a query, these misheard lyrics require phonetic pattern matching techniques to identify the song. We introduce a probabilistic model of mishearing trained on examples of actual misheard lyrics, and develop a phoneme similarity scoring matrix based on this model. We compare this scoring method to simpler pattern matching algorithms on the task of finding the correct lyric from a collection given a misheard query. The probabilistic method significantly outperforms all other methods, finding 5-8% more correct lyrics within the first five hits than the previous best method.

1. INTRODUCTION

Though most Music Information Research (MIR) work on music query and song identification is driven by audio similarity methods, 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 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 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?

Figure 1. Search for misheard lyrics from "Smells Like Teen Spirit" returning results for Guns N' Roses.

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 [1]. However, the problem has only recently been tackled in the MIR community.

Ring and Uitenbogerd [2] 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. [3] developed an acoustic distance metric based on phoneme confusion errors made by a computer speech recognizer. Using this scoring scheme provided a slight improvement over phoneme edit distance; both phonetic methods significantly outperformed a standard text search engine.

In this paper, 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 replicate Ring

and Uitenbogerd'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 outperforms all other techniques, and finds up to 8% more correct lyrics than phonetic edit distance.

2. RELATED WORK

Ring and Uitenbogerd [2] 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 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 [4]. 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 [5].

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. [3] 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.

3. METHOD

3.1 A Scoring Approach

Similar to our method for identifying rhymes in rap lyrics [6], we use a model inspired by protein homology detection techniques, in which proteins are identified as sequences of amino acids. In 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 – those evolving from a shared ancestor [7]. In a BLOSUM matrix M, the score for any two amino acids i and j, is calculated as

$$\mathrm{M}[i,j] = \log_2(\Pr[i,j|\mathrm{H}]/\Pr[i,j|\mathrm{R}]), \qquad (1)$$

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 based on the co-occurrence frequencies of amino acids in alignments of proteins known to be homologous. A positive score indicates a pair is more likely to co-occur in proteins with common ancestry; a negative score indicates the pair is more likely to co-occur randomly. Pairs of proteins with high-scoring aligned regions are labeled homologous.

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 (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.

As for the proteins that give rise to the BLOSUM matrix, these likelihoods are calculated using frequencies of phoneme confusion in actual misheard lyrics. Given a phoneme confusion frequency table F, where $\mathrm{F}_{i,j}$ is the number of times i is heard as j (where j may equal i), the mishearing likelihood is calculated as

$$\Pr[i,j|\mathrm{H}] = \mathrm{F}_{i,j}/\sum_m\sum_n \mathrm{F}_{m,n}. \qquad (2)$$

This corresponds to the proportion of phoneme pairs in which i is heard as j. The match by chance likelihood is calculated as

$$\Pr[i,j|\mathrm{R}] = \mathrm{F}_i \times \mathrm{F}_j/(\sum_m \mathrm{F}_m \times \sum_n \mathrm{F}_n), \qquad (3)$$

where F_i is the total number of times phoneme i appears in the lyrics. This is simply the product of the background frequencies of each phoneme in the pair.

We note that our work is in some ways similar to that of Ristad and Yianilos [8], for learning string edit distance.

3.2 Training Data for the Model

To produce the phoneme confusion frequency table F, we require 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 lyric in the song.

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, traditional songs, and Green Day.

3.3 Producing Transcriptions

We first use 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 specifically for North American English, the language used by the majority of artists in our data [9]. We use the Naval Research Laboratory's text-to-phoneme rules to transcribe any words not found in the dictionary [10].

The transcriptions contain 39 phonemes, consisting of 24 consonants, including affricates such as /tʃ/ and /dʒ/, and 15 vowels, including diphthongs like /ɔɪ/ and /aɪ/ [11]. Additionally, metrical stress is included for the vowels to indicate whether they are part of syllables with primary (1), secondary (2), or no (0) stress. To avoid overfitting due to the relatively small number of secondary stressed syllables in the dictionary, we combine primary and secondary stresses into strong stress to contrast with weak or unstressed syllables. This results in a set of 54 phonemes: 24 consonants and 30 stress-marked vowels.

To better model actual prosody in singing, we reduce 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 introduce an additional symbol for gaps in alignment and treat it like any other phoneme. This would let a "softer" approximant such as /r/ get a lesser penalty when missed than a "harder" affricate such as /tʃ/.

3.4 Iterated Training

We perform an iterated alignment method with the lyric pairs to determine the confusion frequencies. In the first phase, phonemes are 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 results in the highest frequencies for identical phoneme pairs since most of the misheard lyrics contain 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" would be aligned as

ə g ˈɝl w ɪ θ dʒ ˈʌ s t n oʊ h ˈeɪ r
ə g ˈɝl w ɪ θ tʃ ˈɛ s t n ə t h ˈeɪ r,

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

Since gaps do not appear explicitly in any lyrics, we approximate their occurrence by adding gap symbols to the shorter phrase in each pair to ensure the phrases are of the same length. In the example above, we would count one gap, and have it occurring as an /r/ being missed in the F' table. This approximation results in an essentially random initial distribution of gap likelihood across phonemes.

Now, given the initial frequency table, we calculate an initial scoring matrix M' using Equations (1) to (3) above. We then use the scores found in M' to align the pairs in the second phase of training. In this stage, we use dynamic programming to produce the optimum global alignment between each misheard lyric and its corresponding correct version, which may include gaps in each sequence. We then trace back through the alignment and update the phoneme co-occurrences in a new confusion frequency table F. For the example cited above, the new alignment would look like

ə g ˈɝl w ɪ θ dʒ ˈʌ s t n oʊ h ˈeɪ r
ə g ˈɝl w ɪ θ tʃ ˈɛ s t n ə t h ˈeɪ r.

The gap occurs earlier and results in a missed /t/ in the F table. After all the pairs have been processed, we calculate a final scoring matrix M from frequency table F, as above.

3.5 Structure of the Phonetic Confusion Matrix

One interesting property of the phonetic confusion matrix is that, from first principles, we discover perceptual similarities between sounds: if two phonemes a and b have positive scores in our confusion matrix, then they sound similar to the real people who have entered these queries into our database.

Table 1 shows all of the pairs of distinct consonant phonemes a and b such that $M[a, b]$ is positive. These consist 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 are pairs of sibilants (such as /tʃ/ versus /dʒ/ or /ʒ/ versus /ʃ/). All of these similarities are discovered without knowledge of what sounds similar; they are discovered by the training process itself.

When examining stressed vowel 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 differ mainly in height. These pairs include /ɑ/ and /ʌ/, /ʌ/ and /ʊ/, /æ/ and /ɛ/, and /ɛ/ and /ɪ/. Other common confusions include vowels differing mainly in length and diphthongs confused with their

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 1. Non-identical consonants with positive scores.

constituent phonemes, such as /ɪ/ with /i/, /ɑ/ with /aʊ/, and /ɔ/ with /ɔʊ/.

When examining differences in gap scores, we find that the phonemes most likely to be missed (deleted) or heard without being sung (inserted) are /r/, /d/, /ŋ/, and /z/. Although the model is 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. /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 are /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.

3.6 Searching Method

To perform phonetic lyric search with this model, we use matrix M to score semi-local alignments [12] between the query phrase (sequence of phonemes) and all candidate songs in the database. The highest scoring alignment indicates the actual song lyric most likely to be heard as the query, according to our model.

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

$$\mathrm{S}[a,b] = \mathrm{align}(a_o, b_o) + \mathrm{M}[a_v, b_v] + \mathrm{align}(a_e, b_e), \quad (4)$$

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

Searching and training are performed in the same way as with the phonemic method, except that syllables are aligned instead of phonemes. Essentially, this ensures that vowels only match with other vowels and consonants only match with other consonants.

4. EXPERIMENT

To compare the performance of the probabilistic model of mishearing with other pattern matching techniques, we reproduced the experiment of Ring and Uitenbogerd [2] 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.

4.1 Target and Query Sets

We used Ring and Uitenbogerd'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 have corresponding correct lyric phrases that match exactly with a phrase from a single song in the collection. They do 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.

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 Uitenbogerd'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 unavailable.) We removed all test queries from the training set for the probabilistic models.

4.3 Evaluation Metrics

For each method, we found the top 10 best matches for each misheard lyric in our query set and use 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. Thus, if the second returned entry is the correct lyric, we score 0.5 for that query and so on. The hit rate by rank is the cumulative percentage of correct lyrics found at each rank in the results.

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. Mean reciprocal rank after ten results for different search techniques.

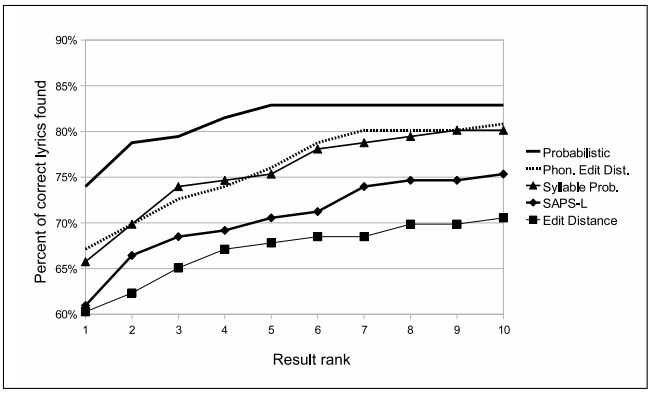

Figure 2. Cumulative percentage of correct lyrics found by rank for different search methods. The probabilistic phoneme model finds 5-8% more correct targets in the first five matches than the next best method. The probabilistic syllable model and phoneme edit distance perform nearly identically, and significantly better than SAPS-L and simple edit distance.

5. RESULTS

The probabilistic model of phoneme mishearing significantly outperformed all other methods in the search task, achieving an MRR of 0.774 and ranking the correct answer for 108 of the 146 queries (74.0%) first. The next best methods were phonemic edit distance and probabilistic syllable alignment, receiving MRRs 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 the t-tests. SAPS-L produced an MRR of 0.655, which was marginally better than the simple edit distance's MRR 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.

The hit rate by rank (Figure 2) further illustrates the effectiveness of the probabilistic phoneme model as it ranks between 5% and 8% more correct lyrics within the top five matches than phonemic edit distance and the probabilistic syllable model. These next two methods appear to perform 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 at finding less similar matches.

5.1 Analysis of Errors

We also observe that the performance of the probabilistic phoneme model plateaus 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.

5.1.1 Differences Among Methods

The phoneme edit distance method did not return any correct lyrics not found by the probabilistic phoneme model. The one query for which SAPS-L returned a hit in the top 10 and the statistical model did not was "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." 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 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 phrases are transcribed without word or syllable boundaries, the only large differences are an inserted /m/ from "Mona" and a missed /b/ from "button."

5.1.2 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;" and "Swingin' the bat" for the Bee Gees' "Stayin' Alive." 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, and "Oh, Lionel (Oh Line')" for Lionel Richie's "All Night Long (all night"). Without knowledge of the song artist, it would be hard to consider these similar to their originals.

The other common problem preventing the algorithms

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 3. Correlation between misheard query length and reciprocal rank of correct answer returned. The positive correlations indicate that longer queries are more likely to have the correct lyric ranked higher, though this effect is least pronounced for the probabilistic phoneme model.

from finding the correct matches for many misheard lyrics stems from the short length of such queries. Some of these include "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 tend to be fairly similar to their correct counterparts, their short length makes it much easier to find exact partial matches which score highly enough to balance the dissimilar remaining portions. Though the models are trained on mishearing, most misheard lyrics tend to have parts heard correctly, so matching identical phonemes will usually give the highest scores. For all 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 3 details these correlations for the different algorithms. Note that this correlation is smallest for the probabilistic phoneme model: it is the least fragile in this way.

5.2 Running Time

The current implementation of the search algorithm is an exhaustive dynamic programming search over the entire collection of lyrics, resulting in O(mn) computing complexity per query, where m is the length of the query and n is the size of the collection. This would likely not be feasible in a commercial application due to the long search time required (about 3 seconds per query on a 1.6 GHz laptop). Xu et al. [3] did demonstrate the effectiveness of using *n*-gram indexing to reduce the running time by pre-computing the distances from 90% of all syllable 3-grams in their collection and pruning off the most dissimilar lyrics. However, this is simpler with Japanese pronunciation than English due to the relatively limited number of possible syllables. Determining the effectiveness of English phoneme *n*-gram indexing while balancing speed, accuracy, and memory use remains an open problem.

6. CONCLUSION

We introduce a probabilistic model of mishearing based on phoneme confusion frequencies calculated from alignments of actual misheard lyrics with their correct counterparts. Using this model's likelihood scores to perform phoneme alignment pattern matching, we were better able to find the correct lyric from a collection given a misheard lyric query. Tested on 146 misheard lyric queries with correct target lyrics in a collection of 2,345 songs, the probabilistic phoneme model produces a Mean Reciprocal Rank of 0.774 and finds up to 8% more correct lyrics than the previous best method, phoneme edit distance, which achieves an MRR of 0.709.

7. ACKNOWLEDGEMENTS

We thank Eric Barberio, from HumorBox Entertainment, for supplying us with the KissThisGuy.com queries we have used in this study. Our research is supported by the Natural Sciences and Engineering Research Council of Canada and by an Early Researcher Award from the Province of Ontario to Daniel Brown.

8. REFERENCES

[1] S. Wright: "The Death of Lady Mondegreen," *Harper's Magazine*, Vol. 209 No. 1254 pp. 48-51, 1954.

[2] N. Ring and A. Uitenbogerd: "Finding 'Lucy in Disguise': The Misheard Lyric Matching Problem," *Proceedings of AIRS 2009*, pp. 157–167, 2009.

[3] X. Xu, M. Naito, T. Kato, and H. Kawai: "Robust and Fast Lyric Search Based on Phonetic Confusion Matrix," *Proceedings ISMIR 2009*, 2009.

[4] R. Gong and T. Chan: "Syllable Alignment: A Novel Model for Phonetic String Search," *IEICE Transactions on Information and Systems*, Vol. 89 No. 1 pp. 332–339, 2006.

[5] D. Norris, J.M. McQueen, and A. Cutler: "Competition and Segmentation in Spoken Word Recognition," *Third International Conference on Spoken Language Processing*, 1994.

[6] H. Hirjee and D.G. Brown: "Automatic Detection of Internal and Imperfect Rhymes in Rap Lyrics," *Proceedings ISMIR 2009*, 2009.

[7] S. Henikoff and J.G. Henikoff: "Amino Acid Substitution Matrices from Protein Blocks" *Proceedings of the NAS*, Vol. 89 No. 22 pp. 10915–10919, 1992.

[8] E.S. Ristad and P.N. Yianilos: "Learning string-edit distance," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 20 No. 5 pp.522-532, 1998.

[9] K. Lenzo: *The CMU Pronouncing Dictionary*, 2007 `http://www.speech.cs.cmu.edu/cgi-bin/cmudict`.

[10] H.S. Elovitz, R.W. Johnson, A. McHugh, J.E. Shore: "Automatic translation of English text to phonetics by means of letter-to-sound rules," *Interim Report Naval Research Lab*. Washington, DC., 1976

[11] International Phonetic Association: *Handbook of the International Phonetic Association: A Guide to the Use of the International Phonetic Alphabet*, Cambridge University Press, 1999.

[12] R. Durbin, S. Eddy, A. Krogh, G. Mitchison: *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids*, Cambridge University Press, 1999.