Published in final edited form as:
J Quant Linguist. 2009 February 1; 16(1): 96–114. doi:10.1080/09296170802514138.

Phonological Distance Measures

Nathan C Sanders and
Department of Linguistics, Indiana University

Steven B Chin[1]
Department of Otolaryngology–Head and Neck Surgery, Indiana University School of Medicine

Abstract

Phonological distance can be measured computationally using formally specified algorithms. This work investigates two such measures, one developed by Nerbonne and Heeringa (1997) based on Levenshtein distance (Levenshtein, 1965) and the other an adaptation of Dunning's (1994) language classifier that uses maximum likelihood distance. These two measures are compared against naïve transcriptions of the speech of pediatric cochlear implant users. The new measure, maximum likelihood distance, correlates highly with Levenshtein distance and naïve transcriptions; results from this corpus are easier to obtain since cochlear implant speech has a lower intelligibility than the usually high intelligibility of the speech of a different dialect.

Phonological Distance Measures

Measuring linguistic distance has proceeded in a number of ways over the last two centuries. Early methods were only applicable to a specific area; in dialectology, for example, Chambers and Trudgill (1998) give the example of drawing dialect boundaries using cognate sets. By the mid-twentieth century, computers were influential enough that they were used to implement some numerical measures, as in the groundbreaking work of Séguy (1973), which began in the 1950's.

This present work continues the recent line of investigation using Levenshtein distance that was begun by Kessler (1995) and is best exemplified by Nerbonne and Heeringa (1997). It also looks to statistical, probabilistic methods that require even less linguistic knowledge and allow even more general applicability. For example, the Levenshtein distance measure used by Nerbonne and Heeringa (1997) allows comparison of any two identical word lists. The probabilistic method developed here is based on a maximum likelihood estimator language classifier explained by Dunning (1994). It allows an estimator trained on an arbitrary input to classify any other corpus, and it can do this with less linguistic knowledge built in to the algorithm.

Both of these measures produce a single scalar which we call "phonological distance." This phonological distance is intended to capture the linguistic knowledge brought to bear by humans when they assess how like or unlike others' speech is to their own. We measure phonological distance between the speech of pediatric cochlear implant users and adult American English speakers who have normal hearing. This measurement results in distances that are easier to measure against a human baseline than previous work because of the relatively low intelligibility between cochlear implant users and naïve listeners. Previous work, such as Gooskens and Heeringa (2004) and Heeringa (2004), has had to work around a possible ceiling effect caused by the high mutual intelligibility of national dialects.

1 Nathan Sanders, 1603 E 3rd St. Apt 28, Bloomington IN 47401 Phone: 812-857-7460; Fax: ncsander@indiana.edu .

In addition, we hope to provide another way to measure progress in cochlear implant user development. These two algorithms each produce a single scalar, similar to existing measures such as the Goldman-Fristoe Test of Articulation (Goldman & Fristoe, 1986). However, these measures produce their results by a fixed algorithm—without any biases or intuitions provided by human calculation except those encoded into the algorithm where they can be examined. Yet the results still correlate well with human perception of intelligibility. Of course, for these algorithms to be usable in the same way as the Goldman-Fristoe Test of Articulation, they would have to be extensively normed over multiple groups.

Dialectology measures the variation of language over an area or space of time (Chambers & Trudgill, 1998). Its quantitative application is known as dialectometry, which began in earnest with the ground-breaking work of Séguy (1973) in determining dialect distances in the French region of Gascony. Indeed, the idea of calling this measure “phonological distance” comes from the different areas of language that Séguy combined to form his overall distance.

Since then, dialectometry has continued to evolve towards methods that minimize the linguistic knowledge required as input. Séguy’s own work was completed after the widespread availability of computers; although the specific phonological characteristics were hand-picked, the method for combining differences to determine distance was mathematically specified. More recently, Nerbonne and Heeringa (1997) used Levenshtein distance, for which the linguistic knowledge necessary is limited to specification of phonological data in terms of phones made of feature bundles, and the assurance that the two different corpora represent the same underlying forms.

Levenshtein distance is used in many fields wherever a measure of similarity between sequences is needed. In bioinformatics, for example, Levenshtein distance is useful for finding similarity between sequences of DNA (Sankoff & Kruskal, 1983). Its wide applicability lies in the fact that it only needs specification of costs between individual items of the sequence. The algorithm specifies how to combine these costs to find the lowest total distance.

Heeringa (2004) gives two specifications for these costs. The earlier proposal, implemented in this paper, uses phonological specification of the costs in terms of number of features changed. The more recent proposal uses phonetic correlates (that is, F1, F2 and F3 measured in Barks) to determine the distance between two segments.

Dunning’s (1994) work on probabilistic language classification provides a starting point for a probabilistic distance measure. Dunning uses a maximum likelihood classifier trained on an n-gram Markov model of language. This produces an estimated likelihood that the training corpus generated the test corpus. He then classifies the language of test corpora by the language of the closest training corpus. This can be viewed as a distance measure by retaining the numerical result and reversing the question asked. Instead of training multiple models, only train one designated as the target language. The likelihood of each test corpus can now be seen as a distance. The reason to prefer such opaque measures is that they obscure, and thus minimize the need for, the knowledge required to obtain the result. This is important to allow the algorithm to be implemented on a computer. We tested both algorithms and compared the results to human judgments of intelligibility.

Methods

Materials

The data analyzed come from three sources: the speech productions of cochlear implant users, a corpus of standard American adult pronunciation, and transcription scores by naïve listeners of the cochlear implant users’ speech productions. The primary data are the productions

collected from pediatric cochlear implant users: they consist of 107 words collected from each child in a picture-naming task designed to provide a relatively complete picture of a developing phonological system and as such to maximize phonological variety (Chin, 2003). The list consists of English monosyllabic words and some disyllables that were created by appending the diminutive suffix /i/ to monosyllables. All word lists were audio-recorded in preparation for transcription.

Two researchers (a clinical linguist and a speech-language pathologist) transcribed children's productions using the International Phonetic Alphabet (International Phonetic Association (IPA), 1999), including the extensions for transcribing disordered speech included as Appendix 3 to IPA (1999). All productions were transcribed independently by the two transcribers and then in consensus. Disagreements were resolved by consensus, with the two transcribers auditing the audio recordings together. All analyses used the consensus transcription.

The comparison corpus is based on the same 107-word list that was obtained from the implant users. It is a baseline meant to represent adult American English speech, as defined by the dictionary pronunciation. In figures, it is referred to as 'base' since that is its primary purpose.

All the transcriptions, both of the cochlear implant users and of the baseline, were digitized. The IPA transcription was encoded using Unicode IPA symbols, and a featural representation was created by stripping diacritics and storing the feature structure in a variant of XML. Since phonological features are used in calculating Levenshtein distance but not in maximum likelihood estimation, diacritics were stripped from the Unicode representations as well to ensure a fair comparison.

Binary features with no weighting were used, with one privative feature of place used for consonants. The features used in consonants and vowels are given below. Features with a constant value for a particular category are labeled as + or −. In particular, all vowels have the features [+approximant +sonorant] in order to provide continuity with sonorant consonants in accordance with the sonority sequencing principle. Using these features, it is less costly to substitute [r] for [i] than it is to substitute [t] for [i] since [r] and [i] have more common feature values.

Features used in consonants

1. approximant
2. consonantal (+)
3. sonorant
4. place (labial, coronal, dorsal)
5. voice
6. strident (coronals only)

Features used in vowels

1. approximant (+)
2. consonantal (−)
3. sonorant (+)
4. back
5. high
6. Advanced Tongue Root (ATR)

7. low
8. round

Finally, transcriptions for each child were collected from naïve listeners. The original recordings were spliced together in a 16-bit 44.1 KHz stereo .wav file with all information in the left channel. In this file, each word was played twice following a numeric prompt. The order was randomized, but used the same random order for all recordings. The listener judges for the transcription were required to have normal hearing, no previous experience with deaf speech, and the ability to write comfortably at the pace of stimulus playback. The judges were recruited from the students at Indiana University. The stimuli were presented to groups of three judges in a sound field using a laptop equipped with external PC speakers. The external speakers were positioned approximately .75 meter distant from the ears of the judges. The judges were told that they would hear English words and their diminutives, but were told nothing about order or whether any words would occur more than once. The judges wrote their transcriptions on a paper form designed for the task. The answers were then tallied for correctness in the same manner as Séguy (1973): mismatches between the transcription and the original words were totaled. The overall score for each child was the mean of three transcription scores from each child's sample, divided by the number of words (107). This produced a number ranging from 0 to 1.

Measures

The materials just described were analyzed according to maximum likelihood distance and Levenshtein distance. The essential difference between the two methods is that maximum likelihood estimation is a statistical measure that combines information from the corpus in a mathematically complex way, but does not provide a complex definition of individual distance. It simply counts the number of times a particular bigram is seen. On the other hand, Levenshtein distance uses a phonologically complex measure of distance between individual phones but has a relatively simple way of combining the distance. This allows its results to be understood easily.

Maximum Likelihood distance

Dunning (1994) gives a method for language classification that uses a simple Markov-based bigram language model. Essentially, it addresses the question "How likely was the training corpus to have generated the test corpus?". The method starts with Bayes' Law in equation 1 and makes the maximum likelihood assumption to obtain the naïve Bayes classifier in equation 2. The naïve Bayes classifier assumes a uniform prior, and assumes that the test data have a probability of one. With the bigram Markov model, the maximum likelihood estimation (MLE) is easily computed as in equation 3.

$$P(\text{training} \mid \text{test}) = \frac{P(\text{test}|\text{training})\, P(\text{training})}{P(\text{test})} \tag{1}$$

$$P(\text{training} \mid \text{test}) = P(\text{test} \mid \text{training}) \tag{2}$$

$$\text{MLE} = \prod_{i=1}^{N} P(\text{bigram}_i) \tag{3}$$

Dunning then chooses the training language that maximizes the likelihood for the test word list. This modification alters only which side of the comparison is being varied: Dunning has only one word list, which is then compared to many possible languages. This method has only one training language, the target, but estimates the distance from multiple test word lists.

For example, assume the following artificial training language "ababcaaaaaa". The bigrams of this language are "ab", "ba", "ab", "bc", … Then the frequency of each bigram can be calculated:

$$\begin{aligned}
&P(\text{``aa''}) = \frac{5 \text{ occurrences}}{10 \text{ bigrams}} = 0.5\\
&P(\text{``ab''}) = \frac{2}{10} = 0.2\\
&P(\text{``ba''}) = \frac{1}{10} = 0.1\\
&P(\text{``bc''}) = \frac{1}{10} = 0.1\\
&P(\text{``ca''}) = \frac{1}{10} = 0.1
\end{aligned}$$

If the estimator is asked to classify the test word "abc", it would be broken into the bigrams "ab" and "bc", giving the likelihood P("ab") · P("bc") = 0.2 · 0.1 = 0.02. In comparison, the string "aab" would have the likelihood P("aa") · P("ab"') = 0.5 · 0.2 = 0.1. "aab" is therefore closer to the language that generated the training corpus, despite the fact that "abc" actually occurs in the training.

Unfortunately, this estimator needs refinement in several areas. The worst problem is that it gives zero probability for any bigram that does not occur in the training, such as "ac" in the example above. This single factor, P("ac")=0, causes the estimated probability of the entire test string to fall to zero—in other words, saying that the distance between the two languages is infinite. This is a real problem when the test corpora consist of phonetically disordered speech, as in our experiments.

The solution is to smooth the input, preventing the appearance of zero probabilities. Smoothing allocates some of the probability space to unseen bigrams. The smoothing method used here is the Good-Turing method, first presented by Good (1953). Good-Turing smoothing estimates the counts of bigrams seen N times from the counts of bigrams seen N+1 times. The precise equation used to determine the expected value for a bigram seen a certain number of times is $r^* = (r+1)(n_{r+1}/n_r$.

For the most interesting case, previously unseen bigrams, r=0 because that is the number of times these bigrams have appeared. Then r+1=1 and n_1 is the number of different bigrams that have occurred only once. Finally, n_0 is the number of bigrams that have never occurred, which can be found by subtracting all bigrams seen any number of times from the total number of possible bigrams. Of course, Good-Turing smoothing is applied to higher numbers as well: for example, the number of bigrams seen six times is estimated from the number of bigrams seen seven times by using the appropriate values of r, n_6, and n_7.

Two minor problems remain. First, longer words give lower likelihoods. To ensure that likelihoods of different test corpora are comparable, we scale the results of each test corpus by the length of the corpus. Second, it is more convenient for both human and computer to use some other measure than raw likelihoods. Since likelihoods are just probabilities, 1.0 means "identical to training language" and 0.0 means "infinite distance from training language". To improve intuitive understanding of the results, we would like a distance measure that is greater for weaker matches between training and test. In addition, storing very small probabilities can be problematic when using hardware-native floating point numbers. Taking the negative logarithm of the likelihoods solves both of these problems: the logarithm converts a 0–1 range

to a 0–−∞ range, which, when negated, produces a number that increases in a way that corresponds nicely with intuitions of distance.

Levenshtein distance

Levenshtein distance was developed by Levenshtein (1965) in order to compare the difference between two sequences. Since it is applicable to any sequence of symbols, it has been used in many fields (Sankoff & Kruskal, 1983). The input to the Levenshtein distance algorithms is a pair of sequences and costs for three operations: insertion, deletion, and substitution. The algorithm then calculates the number of each operation necessary to convert the first sequence to the second. Summing the cost of all operations gives the total distance.

For linguistic distance, Kessler (1995) first used Levenshtein distance on strings of atomic segments. Insertion and deletion costs are each one for atomic characters, so the algorithm is quite transparent: the distance is simply the number of edit operations. For example, the distance between *sick* and *tick* is two—one insertion and one deletion. The distance between *dog* and *dog* is zero, and the distance between *dog* and *cat* is six because all characters must change, so there are three insertions and three deletions. However, there are less obvious comparisons. What is the distance between *realty* and *reality*? What is the distance between *nonrelated* and *words*? (The answers are 1 and 9, respectively). An algorithm to find this distance must have a systematic way to produce the minimum number of necessary changes.

To do this, the Levenshtein algorithm finds and incrementally combines the distances of individual segments in two words. This means that it must check every possible pair of segments, measure the individual distance, and decide which edit operation results in the smallest overall distance at that point. The most efficient method is to store the results in a table using a dynamic programming algorithm. Given a properly structured table, the smallest distance between any sub-sequence is simply the cell at the intersection of the two segments. The minimum distance between the two words is then found at the bottom-right corner of the table.

To state the algorithm more precisely, for any pair of characters s_i and t_j taken from the source word *s* and target word *t*, *levenshtein*(*s*,*t*,*i*,*j*) is *minimum*(*ins*(s_i), *del*(t_j), *sub*(s_i,t_j)). The total distance is *levenshtein*(*s*,*t*,|*s*|,|*t*|).

The functions *ins*, *del*, and *sub* can return arbitrary numbers, but when characters are treated as atomic, the usual definition is

$$ins\left(t_j\right)=1$$
$$del\left(s_i\right)=1$$
$$sub\left(s,t\right)=\begin{cases}0 & \text{if } s_i=t_j\\ 2 & \text{otherwise}\end{cases}$$

In this model, insertion and deletion are the primitive operations, with a cost of one each. Substitution is one insertion and one deletion, giving it a cost of two. However, substitution of a character for itself changes nothing and thus has zero cost. Given these functions, the Levenshtein algorithm will return the minimum number of insertions and deletions necessary for transforming the source to the target.

For example, finding the Levenshtein distance from “ART” to “CAT” creates the table in figure 1. In this table, insertion corresponds to a downward move, deletion to a rightward move, and substitution a diagonal move. The total Levenshtein distance is found at the bottom-right hand corner, but the distances to all intermediate forms are stored in the table as well. The

intermediate form “CATART”, for example, is obtained by inserting three times without any deletions or substitutions. Its cost of 3 is found at the bottom of the first column.

The optimal path is shown in bold. Notice that it follows the diagonal for free substitutions and propagates either down or right in their absence. The final distance is two, indicating that two primitive operators are required; that is, two insertions or deletions. In fact, the table gives them: insert ‘C’ to obtain “CART”, moving down in the table; then delete ‘R’ to obtain “CAT”, moving left in the table. ‘A’ and ‘T’ are common to both words and both produce a diagonal move.

The most direct way to refine Levenshtein distance to take advantage of linguistic knowledge is by changing the definitions of *ins*, *del*, and *sub* to take into account phonetic and phonological properties of segments. When segments are treated as feature bundles instead of merely being atomic, Nerbonne and Heeringa (1997) propose that the substitution distance between two segments simply be the number of features that are different. Two identical segments will therefore have a substitution distance of zero; segments phonetically similar will have a small distance. For example, [k] and [g] would have a distance of one in this system.

Although it increases precision, feature-based substitution causes a number of complications. The first is that substitution distance becomes complicated when not all features are specified for every segment. This is the case, for example, between the vowels and consonants. The minimum difference must be at least the number of unshared features, such as *Advanced Tongue Root* for vowels or *obstruent* for consonants. In other words, the minimum segment distance will always be at least the sum of the non-shared features. The distance of the shared features can then be added on to this baseline. For example, if a consonant with seven features shares only two features with a five-feature vowel, the minimum distance will be eight: $(7 - 2) + (5 - 2) = 5 + 3 = 8$. As a result, the range of distances possible will be a minimum of 8 if all shared features match and a maximum of 10 if none do.

The second complication is obtaining definitions for *ins* and *del* once *sub* is defined. It would be best to retain the original proportions—substitution should cost twice as much as insertion and deletion. To deal with substitution’s variable cost, then, insertion and deletion should be averages. To find the average substitution cost, one can take the average cost of substituting every character for every other character. Then *ins* and *del* return half of this average. With these three functions defined, the table-based algorithm given above can combine feature distances to find the minimum word distance.

Analysis

The maximum likelihood and Levenshtein distances were both applied to the corpora described above. Then they were scaled so as to be comparable to each other and the naïve human transcriptions used as a baseline. Finally, the resulting scores were examined for statistical correlation. The application of the maximum likelihood measure to the children’s data proceeded by training the estimator on the standard dictionary speech. As mentioned above, this process is fairly simple: the algorithm counts the frequency of individual segment bigrams and smooths them appropriately. Next, the maximum likelihood estimator was given the bigrams of each child’s speech and asked to estimate the likelihood. The total distance was obtained and then scaled for length relative to the other corpus from the children.

For Levenshtein distance, there is no training phase, only a standard to which each language is compared. In this case, the standard English word list was the standard and each child’s word list was the test language. This produced 107 individual word scores that were summed to find the total distance.

To correlate the scores obtained for both of these measures with the naïve transcriptions, all scores were normalized to a scale ranging from 0 to 1. The naïve transcription scores need only be divided by 107, the number of words in the list. The other two methods need a definition for minimum and maximum scores: maximum likelihood distance uses the distance of the training corpus to itself as the minimum. The maximum distance is measured with respect to a corpus whose length is the average length of the implant users' corpora. Each segment in this corpus is a never-seen bigram, with no repetitions.

For Levenshtein distance, the minimum distance is 0, and the maximum distance is measured with respect to a corpus whose length is the average length of the implant users' corpora. In addition, each word is the average length and each segment has the average number of features. However, the features are defined to not match with anything, making the cheapest operation always substitution of every feature.

With these maxima and minima defined, all scores can be scaled from 0 to 1. Now it is simple to find the correlation of all three methods using regression analysis. The two mechanical methods are compared to the naïve transcription, and both are compared to each other to find out how well the distances agree.

Results

The results are quite encouraging, both in the comparison to the human baseline, and between the two algorithms. In particular, these experiments lend real weight to the claim that Levenshtein distance accurately models human perceptions. This suggests that Levenshtein distance can profitably be used where appropriate to measure phonological development of implant users.

The raw numeric scores for the naïve human transcriptions are given in table 2. The numbers scaled to a range of 0 to 1 are given in table 3. These results are used as a baseline for comparison with the two computational methods. The corpus codes uniquely identify some cochlear implant user and provide a very rough indication of how long the cochlear implant has been in use. 'base' indicates the baseline, the dictionary pronunciation tested using itself as training.

The raw numeric results for the maximum likelihood estimator are given in table 4 when trained on the corpus 'base', the "dictionary pronunciation". With the base given and a maximum distance calculated as 5.61, the scaled numeric results are given in table 5.

Similarly, the raw numeric scores for Levenshtein distance are given in table 6. Here, the minimum is zero and the maximum is calculated as 680.44. Following in table 7 are the scaled scores.

Finally, regression analysis indicates a correlation of $r=0.925$, $p<0.01$ between Levenshtein distance and naïve human transcriptions. It also found a correlation of $r=0.810$, $p<0.05$ between maximum likelihood distance and human judgments. In addition, there was a correlation between the two algorithms of $r=0.965$, $p<0.001$.

Discussion and Conclusion

Maximum likelihood matches some of the human judgments well, but misses others widely. In particular, the corpora in the middle of the space do worse than those on the end of the scale, indicating (1) that it is only classifying correctly the items on the ends of the scale and (2) that relying on this method to classify all samples will provide too much variation in the middle to be usable.

On the other hand, Levenshtein distance compares favorably with human judgments for nearly every data point. This shows that Levenshtein distance matches naïve human judgments, not just those of experts in dialectology. This is an advancement of Nerbonne and Heeringa's (1997) results, which compared Levenshtein distance only to judgments of expert dialectologists, as well as Gooskens and Heeringa (2004), who compare Levenshtein distance to a survey of perceptibility judgments.

The high statistical correlation between the algorithms and human transcriptions suggest that both algorithms are good at approximating human judgments. Both correlations are at least as great as those found by Gooskens and Heeringa (2004) for correlation of Levenshtein distance to Norwegian dialect judgments.

Conclusion

The two methods investigated by this research provide a single number that measures intelligibility. This "phonological distance" number correlates well with naïve human judgments of intelligibility. Levenshtein distance, especially, is well enough correlated to be usable in place of an intelligibility measurement.

Since both methods are algorithmic and implementable on a computer, they produce their results automatically once they are given transcriptions as input. Although their respective models of distance are impoverished compared to actual human representations, analysis of their performance and judgments can provide some insight into aspects of human processing.

It is interesting that the results of the two methods are so highly correlated. They seem to be capturing similar aspects of distance, despite their differences in its definition. It could be that frequency alone correlates quite well with human judgments, implying that a large component of intelligibility can be traced to simple knowledge of probability. On the other hand, since the feature structure used in Levenshtein distance further improves performance, one might also conclude that phonology models something useful about intelligibility.

On a practical note, adapting the statistical method of maximum likelihood estimation gives an algorithm that has the usual strengths of statistical algorithms. It requires few decisions about linguistic matters and can be trained on a variety of corpora, not just parallel transcriptions of the same text. Finding that its performance is similar to Levenshtein distance is encouraging. However, the present scarcity of phonetically transcribed corpora makes its flexibility less useful.

Furthermore, Levenshtein distance is surprisingly easy to alter due to its flexible design: feature-based knowledge integrates easily into the basic algorithm. It is not so obvious how the probabilistic algorithm could be extended to model phonological features.

This work extends previous work in two ways: first, comparison to human judgments extends the work of Nerbonne and Heeringa by showing that Levenshtein distance reproduces intelligibility judgments as measured by naïve listener transcription. Previous work used as standards judgments of expert dialectologists (Nerbonne & Heeringa, 1997) and self-reported judgments of dialect similarity (Gooskens & Heeringa, 2004). In particular, the Levenshtein distance of cochlear implant users' speech correlates even more highly with human judgments than did the previous work on dialects. This is likely a consequence of a better measuring method, naïve listener transcription, that was unavailable to Gooskens and Heeringa because the high mutual intelligibility of Norwegian dialects would cause a ceiling effect.

Second, this work compares Levenshtein distance with a novel application for maximum likelihood estimation that is adapted from Dunning's (1994) language classification. This

method shows that the same results can be obtained using even less linguistic knowledge, and can, with appropriate adaptation, be used to test refinements of dialect distance by implementing changes in both algorithms.

Future Work

Later refinements by Nerbonne and Heeringa (2001) showed that weighting features by information gain can improve the results of Levenshtein distance. It might be advantageous to investigate feature weighting or non-binary features in order to increase phonetic accuracy. Gooskens and Heeringa (2004) and Heeringa (2004) use phonetic correlates to determine Levenshtein edit costs. Unfortunately, this throws out phonological abstraction, which a real featural system retains.

It would be interesting to compare a version of Levenshtein distance that has no knowledge of feature structure with the maximum likelihood distance, or to find a way to make the maximum likelihood estimator aware of feature structure.

Kondrak's (2002) algorithm differs from Nerbonne's in two respects for distance: it incorporates the notion of phonetic coalescence and break-up, and it uses a non-binary feature system. Kondrak does not compare his algorithm to Nerbonne's because the applications are different; it would be interesting to see if these alterations would improve the performance of the Levenshtein distance.

Acknowledgments

The project described was supported by Grant Number R01DC005594 from the National Institutes of Health to Indiana University. Its contents are solely the responsibility of the authors and do not necessarily represent the official views of the National Institutes of Health. We would also like to thank Cara Lento Kaiser and Amy P. Teoh for assistance with phonetic transcriptions, and Andrew K. Kirk and Jason K. An for assistance with naïve listener transcriptions.

References

Chambers, JK.; Trudgill, P. Dialectology. Second ed. Cambridge University Press; Cambridge, United Kingdom: 1998.

Chin SB. Children's consonant inventories after extended cochlear implant use. Journal of Speech, Language, and Hearing Research 2003;46:849–862.

Dunning, T. Statistical identification of language. New Mexico State University; 1994. (Tech. Rep.)

Goldman, R.; Fristoe, M. Goldman-Fristoe test of articulation. American Guidance Service; Circle Pines, MN: 1986.

Good IJ. The population frequencies of species and the estimation of population parameters. Biometrika 1953;40:237–264.

Gooskens CS, Heeringa WJ. Perceptive evaluations of Levenshtein dialect distance measurements using Norwegian dialect data. Language Variation and Change 2004;16(3):189–207.

Heeringa, WJ. Doctoral dissertation. University of Groningen; 2004. Measuring dialect pronunciation differences using Levenshtein distance.

International Phonetic Association (IPA). Handbook of the International Phonetic Association: A guide to the use of the International Phonetic Alphabet. Cambridge University Press; Cambridge, UK: 1999.

Kessler, B. Computational dialectology in Irish Gaelic. Proceedings of the European ACL; Dublin. 1995. p. 60-67.

Kondrak G. Algorithms for language reconstruction (Doctoral dissertation, University of Toronto). Dissertation Abstracts International 2002;63(12):5934.

Levenshtein VI. Binary codes capable of correcting deletions, insertions, and reversals. Doklady Akademii 1965;163(4):845–848.

Nerbonne, J.; Heeringa, WJ. Measuring dialect distance phonetically. In: Coleman, J., editor. Workshop on computational phonology. Madrid: 1997. p. 11-18.

Nerbonne J, Heeringa WJ. Computational comparison and classification of dialects. Dialectologia et Geolinguistica 2001:69–83.

Sankoff, D.; Kruskal, JB. Time warps, string edits, and macromolecules: The theory and practice of sequence comparison. Addison-Wesley; Reading MA: 1983.

Séguy J. La dialectometrie dans l'atlas linguistique de la gascogne. Revue de linguistique romane 1973;37:1–24.

		A	R	T
	0	1	2	3
C	**1**	2	3	4
A	2	**1**	**2**	3
T	3	2	3	**2**

Figure 1.
The distance table for “ART” to “CAT”

Table 2

Baseline of naïve human transcriptions

Corpus	Distance
sgl20	32.67
sif20	41
siz20	45.67
sgj20	27
siw20	54.33
see26	66.33
sgb20	82

Table 3

Baseline of naïve human transcriptions, scaled

Corpus	Distance
sgl20	.3053
sif20	.3852
siz20	.4268
sgj20	.2523
siw20	.5078
see26	.6199
sgb20	.7664

Table 4

Maximum likelihood estimator trained on Base

Corpus	Distance
base	2.10
sgl20	2.34
sif20	2.31
siz20	2.36
sgj20	2.40
siw20	2.36
see26	2.53
sgb20	2.57

Table 5

Maximum likelihood estimator trained on Base, scaled

Corpus	Distance
sgl20	.1623
sif20	.1402
siz20	.1771
sgj20	.2073
siw20	.1785
see26	.2935
sgb20	.3252

Table 6

Feature-based Levenshtein distance

Corpus	Distance
base	0
sgl20	250.2
sif20	258.9
siz20	317.0
sgj20	318.9
siw20	380.0
see26	545.6
sgb20	630.0

Table 7

Feature-based Levenshtein distance, scaled

Corpus	Distance
sgl20	.3480
sif20	.3600
siz20	.4407
sgj20	.4447
siw20	.5284
see26	.7587
sgb20	.8761