IMPROVED ESTIMATION OF PROBABILITIES IN PRONUNCIATION BY ANALOGY

Janne V. Kujala[1], Asoke K. Nandi[1,2]

[1]Department of Mathematical Information Technology, University of Jyväskylä, Finland
[2]Department of Electrical Engineering and Electronics, University of Liverpool, UK
email: jvk@iki.fi, a.nandi@liverpool.ac.uk

ABSTRACT

Pronunciation by Analogy is a method for generating phonetic transcriptions for previously unseen written words based on matching substrings of known words and their pronunciations. The method inherently generates several candidate pronunciations and a multitude of heuristics have been proposed for choosing the best one. In [1], a theoretically justified probabilistic approach for scoring the pronunciations was proposed, with performance on par with the best heuristic methods. However, a certain ad hoc modification—a fractional power applied to the estimated probabilities of the substring pronunciations—was also found to improve performance. In this article, we give an explanation for this unexpected improvement. We show that the fractional power in fact improves the estimates of the candidate pronunciation probabilities. This also gives an indirect explanation of the good performance of the current best heuristic proposed in [2].

Index Terms— Pronunciation by Analogy, probability estimation, Bayesian model averaging.

1. INTRODUCTION

Generating the pronunciation for an unseen word is a difficult problem for certain languages, notably English. There are several different approaches for the task but Pronunciation by Analogy (PbA) is among the best and works surprisingly well given its relative simplicity. PbA was originally proposed by Glushko in 1979 [3] as a psychological model of how humans pronounce pseudowords; a concrete algorithm was given by Dedina and Nusbaum [4] with their PRONOUNCE program. The method inherently generates several candidate pronunciations and several heuristics have been proposed to choose the best one among them. The most successful implementation of such heuristics was formulated by Marchand and Damper [5] in 2000, who presented five heuristic strategies and a method from information fusion literature to combine the scores of these into a combined strategy [5, 6]. Further improvements were recently made by Polyákova and Bonafonte [2] who proposed another six component strategies, one of which (Strategy 11) turned out to be very good.

Since then, Kujala and Keurulainen [1] proposed an alternative, probabilistically justified method for scoring the different candidate pronunciations. This new method performs as well or even better than any of the heuristic methods. The good performance of the current best performing heuristic, Strategy 11 of Polyákova and Bonafonte [2], can be explained by its similarities to the theoretically justified probabilistic method. However, one apparently arbitrary aspect of Strategy 11 remained unexplained and it turns out [1] that a variant of the same idea can be used to improve the performance of the probabilistic method as well. In the present paper, we explain this unexpected result. We show that the seemingly ad hoc modification works precisely because it accounts for certain non-idealities in the theoretically justified algorithm.

2. PRONUNCIATION BY ANALOGY

Suppose we have a dictionary consisting of aligned pairs (x, y) of words and pronunciations so that every letter of x correspond to one phoneme at the same position in y. Obviously such an alignment requires silent phonemes to be inserted in the pronunciations (among other technicalities), but this is the type of training data that is used in most PbA work. Most work uses the 20,009 word NETtalk corpus, which is manually aligned by Sejnowski and Rosenberg [7].

The gist of a PbA method is as follows.[1] As a preprocessing, for every substring appearing in any word of the training corpus, the frequency of each of its possible pronunciations over the whole corpus is calculated. Then, a pronunciation for a new input word is generated by considering different segmentations (typically all segmentations with the minimum number of segments) of the input word into substrings and concatenating the previously seen pronunciations of these substrings. This process inherently yields several candidate pronunciations and different variants of PbA differ most notably in how they choose the best one among the candidates (usually basing the choice somehow on the frequency data).

[1]We are giving a simplistic view here, ignoring certain details which are not important for the present context (such as the use overlapping segments, special handling of the beginning and end of words, etc., see [1] for the details); none of these, however, are essential for the present focus although they are certainly important in their own right.

The accuracy of a PbA algorithm is usually quantified by so called leave-one-out evaluation, in which each word in turn is temporarily removed from the training corpus and it is tested whether the removed word can be pronounced correctly by analogy with the remaining words in the corpus. The overall accuracy of the algorithm is the proportion of words that were correctly pronounced. We apply statistical testing as in [5] to determine if an increase in accuracy from a given baseline result is statistically significant.

2.1. Probabilistic approach

The probabilistic approach proposed in [1], in its simplest form, works as follows.

For every segmentation $x = x_1 + \cdots + x_n$ of the input word x into segments x_i, $i = 1, \ldots, n$, estimate the probability of every candidate pronunciation $y = y_1 + \cdots + y_n$ as

$$\hat{p}(y_1 + \cdots + y_n \mid x_1 + \cdots + x_n) = \prod_{i=1}^{n} \frac{(\text{\# of times } x_i \text{ is pronounced as } y_i \text{ in the corpus})}{(\text{\# of times } x_i \text{ appears in the corpus}) + 1}. \quad (1)$$

The $+1$ at the denominator makes the probability estimation more accurate by accounting for the possibility that a segment might have a previously unseen pronunciation.

After the probabilities $\hat{p}(y_1 + \cdots + y_n \mid x_1 + \cdots + x_n)$ given a segmentation have been estimated, these probability distributions are then averaged over all different segmentations $x = x_1 + \cdots + x_n$ of the input word with the minimum number n of segments. The candidate pronunciation y with the highest averaged probability is then output as the most likely pronunciation of the input word. This corresponds to Bayesian model averaging and is mathematically optimal assuming that one of the considered segmentations is the correct generating model (and that a priori, any of the models is equally likely to be the correct model).

For simplicity, we have only considered non-overlapping segments here as in the method of Sullivan and Damper [8]. However, it should be noted that the probabilistic method we consider here has been generalized to overlapping segments in [1] and our new results do generalize to that case as well.

2.2. Strategy 11 of Polyákova and Bonafonte [2]

The best performing current heuristic, Strategy 11 of Polyákova and Bonafonte [2] is essentially as follows. Each candidate pronunciation is scored as

$$\sum \left[\prod_{i=1}^{n} (\text{\# of times } x_i \text{ is pronounced as } y_i \text{ in the corpus}) \right]^{1/n}, \quad (2)$$

where the sum is over all segmentations with the minimum number n of segments, and the pronunciation y with the highest sum is output as the best candidate. Obviously the summing is equivalent to the model averaging of the probabilistic method so the only differences are that Strategy 11 uses raw frequencies rather than the normalized frequencies of the probabilistic approach and Strategy 11 computes the geometric mean rather than the product of the values of the segments.

In [1] it is shown that the estimated probabilities (which are theoretically justified) are generally better than using raw frequencies (the product of which can be considered an ad hoc function). However, it turns out that the geometric mean, which differs from the product by the n-th root operation, in fact clearly improves performance. Thus, we are faced with the dilemma that while every other probabilistically justified modification to the heuristic methods improved performance, the removal of the seemingly ad hoc root function decreased performance.

This issue was considered in [1] and it was shown there that a constant root is generally better than the varying n-th degree root. Thus, the n-th root works only because n, the number of segments, happens to generally be near the optimal constant value of n; it is the shape of the root function itself that is important. However, the surprising fact remains that the application of a "magic function" to the estimated probabilities before model averaging improves performance in the otherwise theoretically justified probabilistic algorithm.

In the following, we show that the root function in fact makes the estimates of the probabilities more accurate and so there is a direct explanation of the improved performance.

3. ESTIMATION OF PROBABILITIES

On an abstract level, the probabilistic PbA algorithm considers several potentially correct models m (segmentations), each generating an estimated probability distribution $\hat{p}(y \mid x, m)$ over different candidate pronunciations y for the input word x. Then, these distributions are averaged over the prior distribution $p(m)$ of the models to yield the final estimate of the probability of correctness

$$\hat{p}(y \mid x) = \sum p(m)\hat{p}(y \mid x, m) \quad (3)$$

for each candidate pronunciation y. Now, it turns out that in practice, a better result is obtained by using instead the estimate

$$\hat{p}(y \mid x) = \sum p(m)\hat{p}(y \mid x, m)^{\alpha} \quad (4)$$

for some $0 < \alpha < 1$ (where $\alpha = 1/(\text{degree of the root})$).

The fact that it is a power function that works so well as a "magic function" is not surprising as it is unique in that it has the property that it can be applied to a product factorwise (the probabilities $\hat{p}(y \mid x, m)$ are products of the probabilities of segments) and it also preserves the range $[0, 1]$ of probabilities.

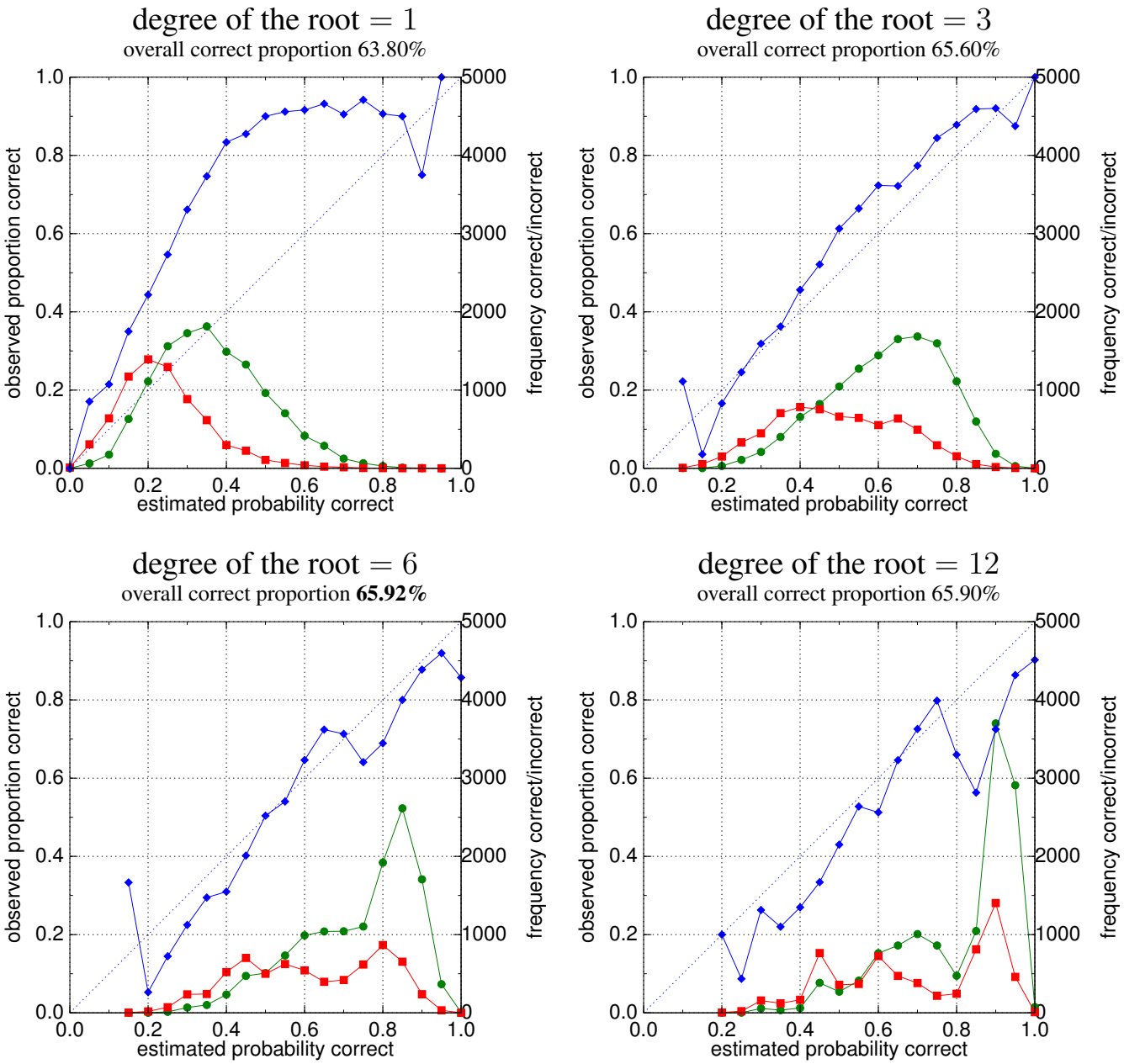

Fig. 1. A plot of the estimated probability of correctness of the candidate pronunciation versus the observed proportion of correct outputs for different roots applied to the estimated probabilities before model averaging in the probabilistic PbA algorithm described in the text (for NETtalk corpus, speech-to-text direction). The circles (green) give the frequency for correct outputs, the squares (red) give the frequency for incorrect outputs, and the diamonds (blue) give the proportion of correct outputs. Generally the overall correct proportion appears to be highest where the estimated probabilities are closest to the observed correct proportions (around degree 6).

However, what we are trying to clarify here is the mechanism by which this increases the actual probability of correctness of the output pronunciation

$$y^* = \arg\max_y \hat{p}(y \mid x). \tag{5}$$

The answer can be seen in Figure 1. The root brings the estimated probability $\hat{p}(y \mid x)$ closer to the true observed proportion, and hence makes the choice of the best candidate more accurate. The agreement of the true and estimated probabilities obtained by the root is surprisingly good.

There was no reason to expect such a good agreement as the choice function is insensitive to any monotone transfor-

NETtalk text-to-speech

Root	Distance	Accuracy (%)
1	0.2955	63.80
2	0.2183	65.25
3	0.2000	65.60
4	**0.1967**	65.81
5	0.1982	65.87
6	0.2011	**65.92**
7	0.2043	65.89
8	0.2076	65.88
12	0.2181	65.90
24	0.2338	65.87
48	0.2441	65.87
$\to \infty$	0.2562	65.85
"∞"		60.53

NETtalk speech-to-text

Root	Distance	Accuracy (%)
1	0.2528	76.23
2	0.1714	76.66
3	0.1545	**76.83**
4	**0.1518**	76.74
5	0.1531	76.69
6	0.1558	76.65
7	0.1587	76.58
8	0.1615	76.51
12	0.1701	76.51
24	0.1825	76.50
48	0.1905	76.49
$\to \infty$	0.1998	76.49
"∞"		71.32

CMUDict speech-to-text

Root	Distance	Accuracy (%)
1	0.2939	71.82
2	0.1999	72.80
3	0.1749	**72.91**
4	0.1684	72.86
5	**0.1682**	72.75
6	0.1703	72.66
7	0.1731	72.58
8	0.1761	72.53
12	0.1863	72.43
24	0.2021	72.39
48	0.2125	72.38
$\to \infty$	0.2251	72.38
"∞"		66.74

CMUDict speech-to-text

Root	Distance	Accuracy (%)
1	0.2506	77.47
2	0.1689	78.16
3	0.1495	**78.30**
4	**0.1452**	78.30
5	0.1455	78.25
6	0.1474	78.18
7	0.1498	78.14
8	0.1523	78.10
12	0.1605	78.02
24	0.1729	77.98
48	0.1810	77.98
$\to \infty$	0.1906	77.98
"∞"		73.21

Table 1. Evaluation of the probabilistic algorithm for different values of the root. The distance value quantifies the difference between the estimated probability of correctness of the candidate pronunciation and the observed probability of correctness (see the text for details). The infinite root corresponds to Strategy 3 of Marchand and Damper [5]. The standard error of the accuracy (%) scores is around 0.3.

mation of the final probabilities $\hat{p}(y \mid x)$. Thus, the same improved overall performance of the algorithm could logically have been obtained for any monotone correspondence between the estimated and true probability of correctness of y.

The same qualitative results generally hold for different data sets and different variants of the algorithm proposed in [1]. For example, if we repeat the same experiment for the 112,102 word automatically aligned CMUDict corpus (obtained from `http://www.pascal-network.org/Challenges/PRONALSYL/`) or consider the speech-to-text direction (obtained by swapping the written words with their pronunciations) for either corpus we obtain visually the same pattern as shown in Figure 1.

To measure the agreement between the true and estimated probabilities objectively, we quantify it using the square distance

$$\frac{\sum_i N_i (p^i_{\text{est}} - p^i_{\text{obs}})^2}{\sum_i N_i}, \tag{6}$$

where N_i is the number of cases (sum of the red and green points in Figure 1) corresponding to a given estimated probability p^i_{est} and p^i_{obs} is the actual observed proportion of correct outputs in these cases. In other words, this is the average squared distance between the blue curve and the diagonal in Figure 1 weighted by the proportion of cases.

Table 1 shows this distance and the overall correct proportion (accuracy) for different values of the root for the four data sets. In all cases, the accuracy is best around the smallest values of the distance. The increase in accuracy from the baseline result (root $= 1$) is statistically very significant ($z > 4.2$, $p < .000013$) in all cases with larger roots except for the NETtalk speech-to-text direction where only the roots of de-

gree 3 ($z = 2.0$, $p = .024$) and 4 ($z = 1.7$, $p = .047$) yield statistically significant increases.

Another feature worth noting of the data sampled in Table 1 is that even when the degree of the root increases above the optimal values and towards infinity, the performance of the algorithm does not drop too much but appears to asymptote slightly below the optimal performance. This is unexpected at a first glance because for $\alpha = 0$ in Eq. (4) (corresponding to infinite root), the decision function is in fact equivalent to that of Strategy 3 (the best performing of the original 5 strategies proposed by Marchand and Damper [5]), which performs much worse than the probabilistic method with $\alpha = 1$ in this case. Strategy 3 scores each candidate pronunciation simply as the number of times it appears as a possible pronunciation over the different segmentations.

However, although the estimated probabilities change continuously for $\alpha \in [0, 1]$, the decision function itself changes abruptly at 0. An exponent close to 0 maps every probability to approximately 1.0 and so the top candidates are always those corresponding to the maximum number of the same pronunciation, as in Strategy 3. However, the small differences of the probabilities still have an effect in ordering the otherwise tied candidates. It can be shown that at the limit $\alpha \to 0^+$, this order is given by the product (rather than the average) of the estimated probabilities of the candidates with the same pronunciation. Thus, while the performance at degree ∞ of the root is 60.53% (corresponding to Strategy 3), the performance tends to 65.85% as $\alpha \to 0^+$.

4. CONCLUSION

We have shown that applying a root function (with optimal degree typically around 2–6) to the estimated probabilities of segment pronunciations improves the accuracy of the estimated probability of the whole pronunciation obtained by averaging the product of segment probabilities over different segmentations.

This explains why the geometric mean applied in the heuristic Strategy 11 of Polyákova and Bonafonte [2] works so well—the geometric mean applied therein corresponds to an n-th root, where n, the number of segments, often happens to be within the optimal range of the degree of the root. Thus, extending on [1], we have now shown that the current best heuristic, Strategy 11, differs from the probabilistic method of [1] by a series of small changes, each of which, when applied to Strategy 11, categorically improves its performance and brings it closer to a probabilistically consistent formulation. In numbers, the improvement is from 64.00% of Strategy 11 (or from 66.14% if using information fusion methods to yield a best combination of all 11 component strategies) to 66.61% of the best variant of the probabilistic methods of [1]. This difference of 2.61% in accuracy is equivalent to the correct pronunciation of an additional 511 words, arising from the improved estimation of probabilities as explained here. The threshold for statistical significance at the 99% confidence level is 207 words, so this improvement is clearly statistically significant ($z = 7.6$, $p < 10^{-13}$).

The degree of the root has been a kind of arbitrary tuning parameter in the otherwise theoretically justified method in [1]. However, in the present paper, we have shown a principled basis for choosing its value: it should be chosen so as to make the probability estimates as accurate as possible, that is, to yield a straight, diagonal line in Figure 1. The question is still open why the estimated probabilities were systematically skewed in the first place, but now we know that the root function quite effectively accounts for this non-ideality in the method.

5. ACKNOWLEDGMENTS

This work was financially supported by TEKES (Finland) grant 40334/10 'Machine Learning for Future Music and Learning Technologies'.

6. REFERENCES

[1] Janne V. Kujala and Aleksi Keurulainen, "A probabilistic approach to pronunciation by analogy," *arXiv:1109.4531v1*, 2011.

[2] Tatyana Polyákova and Antonio Bonafonte, "New strategies for pronunciation by analogy," in *Proceedings of ICASSP '09*, 2009, pp. 4261–4264.

[3] Robert J. Glushko, "The organization and activation of orthographic knowledge in reading aloud," *Journal of Experimental Psychology: Human Perception and Performance*, vol. 5, pp. 674–691, 1979.

[4] Michael J. Dedina and Howard C. Nusbaum, "PRONOUNCE: a program for pronunciation by analogy," *Computer Speech and Language*, vol. 5, pp. 55–64, 1991.

[5] Yannick Marchand and Robert I. Damper, "A multistrategy approach to improving pronunciation by analogy," *Computational Linguistics*, vol. 26, no. 2, pp. 195–219, 2000.

[6] R. I. Damper and Y. Marchand, "Information fusion approaches to the automatic pronunciation of print by analogy," *Information Fusion*, vol. 7, pp. 207–230, 2006.

[7] Terrence Sejnowski and Charles Rosenberg, "Parallel networks that learn to pronounce english text," *Complex Systems*, vol. 1, pp. 145–168, 1987.

[8] K. P. H. Sullivan and R. I. Damper, "Novel-word pronunciation: a cross-language study," *Speech Communication*, vol. 13, no. 3–4, pp. 441–452, 1993.