Statistical Evaluation of Pronunciation Encoding

Iris Merkus, Florian Schiel

Bavarian Archive for Speech Signals, Ludwig-Maximilians-Universität München, Germany
Schellingstr. 3, 80803 München, Germany
iris@bas.uni-muenchen.de, schiel@bas.uni-muenchen.de

Abstract

In this study we investigate the idea to automatically evaluate newly created pronunciation encodings for being correct or containing a potential error. Using a cascaded triphone detector and phonotactical n-gram modeling with an optimal Bayesian threshold we classify unknown pronunciation transcripts into the classes 'probably faulty' or 'probably correct'. Transcripts tagged 'probably faulty' are forwarded to a manual inspection performed by an expert, while encodings tagged 'probably correct' are passed without further inspection. An evaluation of the new method on the German PHONOLEX lexical resource shows that with a tolerable error margin of approximately 3% faulty transcriptions a major reduction in work effort during the production of a new lexical resource can be achieved.

Keywords: pronunciation dictionary, error detection, phonotactical model

1. Introduction

Applications of automatic speech recognition and automatic speech synthesis (among others) require some form of encoding to represent canonical or likely pronunciations of words. In the simplest and most common form, the pronunciation of words is encoded by a linear string of phonetic symbols drawn from a standardized phoneme set for the particular language (e.g. IPA (International Phonetic Association, 1999), SAM-PA (Wells, 1997)). Other more sophisticated encodings feature multiple possible pronunciations or a finite state automaton per word form to cover dialectal and speaker-individual variability. In most cases these so called technical pronunciation dictionaries contain full word forms, where each lemma of the language is coded into a variety of derived word forms separately. Also, for pragmatic reasons technical pronunciation dictionaries often contain additional words of foreign languages (nowadays mostly adopted English terms) and proper names like person names, street names, city names etc. (Jurafski and Martin, 2009). Many technical and scientific projects require the production of a tailored technical pronunciation dictionary. Also, most speech corpora feature such a dictionary as part of the corpus to cover all words in the recorded data. Insofar the task of creating such a dictionary is a frequently encountered and common task in language technology and science.

Since most languages do not have a one-to-one mapping between lexemes and phonemes, this task cannot be automated satisfactorily. During the last two decades many so called text-to-phoneme (TTP) systems have been developed for a large number of common world languages (see for instance the ECESS project for European languages[1]) and evaluated in benchmarks against manually corrected pronunciation encodings by Phoneticians. Unfortunately almost no TTP systems are freely available (with some exceptions). Rule based and stochastic approaches have been studied and applied mainly within systems for automatic speech synthesis. Depending on the language, on the domain (task) and on the quality of the approach error rates between 5 and 20% are typical (e.g. (Reichel and Schiel, 2005)). To obtain better error margins pronunciation dictionaries have to be corrected by experts manually, at best in form of a multiple-pass annotation with some quality assessment schema (e.g. majority vote) and the possibility to assess the quality of the resulting dictionary by calculating the inter-labeler agreement (e.g. Cohen's kappa (Cohen, 1960)). Consequently the production of a large dictionary requires considerable time and resources, which are often not available.

To our knowledge there exists no literature dedicated to assist the process of encoding pronunciation by automatic means. There exists recommendations for the formal description of lexical information[2], language specific guidelines for the proper and consistent use of a phonetic symbol inventory[3], and there are several project-internal guidelines and reports on how to deal with the manual encoding process (e.g. TC-STAR, Verbmobil) but no serious attempts to speed up the encoding process.

At the Bavarian Archive of Speech Signals (BAS)[4] every speech corpus is required to contain a technical pronunciation dictionary; speech corpora lacking this resource are not considered for distribution, since the pronunciation dictionary is a major factor to pass the BAS internal validation protocol[5]. Although the majority of word forms in a newly created dictionary can be recycled from other, already corrected dictionaries (e.g. the PHONOLEX[6] core list for German), there remain about 30-40% (depending on the domain of the speech corpus) of previously 'un-seen' word

[1] European Center of Excellence in SpeechSynthesis, www.ecess.eu/

[2] e.g. the Text Encoding Initiative, P5 Guide Lines, Chapter 9; www.tei-c.org/release/doc/tei-p5-doc/en/html/DI.html

[3] e.g. for German the BAS 'Transcription Conventions for Canonical German' by Sonja Biersack; www.bas.uni-muenchen.de/Bas/BasGermanPronunciation/

[4] www.bas.uni-muenchen.de/Bas

[5] see www.phonetik.uni-muenchen.de/forschung/BITS/ Revalidierungen.html for examples of BAS revalidations

[6] www.bas.uni-muenchen.de/Bas/BasPHONOLEXeng.html

forms that need to be coded from scratch. We estimate that the production of a canonical, linear pronunciation dictionary with tolerable error margins for a speech corpus requires about 2-5% of the total corpus production costs. To alleviate this process it would be helpful to identify error-prone newly created encodings by automatic means and then restrict the manual correction to these.

This paper describes a new method to achieve this automatic filtering of newly created pronunciation encodings for the manual correction process of a technical pronunciation dictionary. The aim is to find a fully automatic method to isolate those entries in a newly created technical pronunciation dictionary which have to be inspected by an expert. The remaining new encodings are passed to the final resource without inspection. In this study we are concerned with linear canonical pronunciation coding only; other encodings (for instance graphs) may be treated in analogy. In the following we describe the method and evaluate it for German on parts of the PHONOLEX dictionary to assess the quality of the new method.

2. Automatic Detection of Coding Errors

The basic idea is to first create a pronunciation encoding by some automatic method (e.g. a TTP algorithm) and then automatically flag a pronunciation for possible transcription errors when the phonotactics of the encoding is unlikely for the particular language, in our case German. The flagged encodings are then passed to an expert for inspection and correction before inserting them into the final dictionary.

A simple and straightforward way to put this into practice would be the usage of several independent TTP algorithms running in parallel. Then a majority-vote or other merging technique could be used to flag those orthographic entries for which the encoding has to be inspected manually. Unfortunately for most languages (including German) there are no multiple TTP algorithms available. Therefore in this study we pursue the matter by means of a statistical model.

2.1. Statistical Model for Phonotactics

Phonotactics can be expressed by the occurrence and non-occurrence of certain tuples of phonetic symbols within a pronunciation encoding of spoken texts. To estimate the probability for a correct phonotactics we therefore calculate the normalized log tri-gram probability $L(A, M)$ of a string of phonetic symbols $A = s_1 \dots s_N$ with regard to a tri-gram model M as the sum over the log bigram from a virtual word initial marker '#' to the first phonetic symbol in the encoding (the uni-gram of word initials) and all possible log tri-gram probabilities for the remaining sequence. To compensate for different string lengths we normalize the total sum by the number of symbols in the word N:

$$\begin{aligned} L(A, M) &= \frac{1}{N} log(P(s_1|s_0)) + \\ &+ \frac{1}{N} \sum_{n=2}^{N} log(P(s_n|s_{n-2}, s_{n-1})) \end{aligned}$$

with $s_0 =$ '#'. So, basically L is the mean log likelihood per phonetic symbol in a transcription A and is therefore comparable to L of other phonetic transcriptions based on the same model M. In the following we train tri-gram models to different pronunciation lists with correct and (partially) faulty encodings. The hypothesis is that unknown but correctly encoded pronunciations will produce higher probability estimates on a tri-gram model trained on correct transcriptions while an unknown faulty encoding will produce higher estimates on a tri-gram model trained on (partially) faulty transcriptions.

2.2. Database

Although the described method can be applied to any language, we'll describe the process in this paper for a concrete application in German. The application to other languages requires a matching data base of pronunciation lists as described in the following.

All pronunciation lists used in this study are coded in German SAM-PA (Wells, 1997) and drawn from the German PHONOLEX project (Schiel et al, 1999). PHONOLEX consists of two pronunciation lists: the CoreList which contains only pronunciation encodings inspected by expert Phoneticians and are assumed to be correct, and the PhonList which consists of encodings from different other sources, mostly TTP algorithms or hand-crafted encodings without quality control. Since these have not been inspected by the same expert team, they are assumed to contain encoding errors. In an earlier study (Libossek and Schiel, 2000) 17.9% pronunciation encoding errors in a lexicographic-representative sample of 4685 entries drawn from PhonList were found. If we extrapolate this finding to the whole PhonList, we can estimate that roughly every 5th entry in PhonList is faulty. For the purpose of this study a list with exclusively faulty, realistic pronunciation encodings would have been optimal, but unfortunately such a resource was not available in sufficient size. Therefore we use the CoreList as the empirical basis for correct pronunciations and the PhonList as the best approximation for faulty pronunciations.

As a first step both lists were filtered for double orthographic entries, errors, word fragments etc. resulting in 59.546 entries for CoreList and 992.476 entries for PhonList. The two lists have a common set of 41.521 orthographic entries (EqualList). To achieve maximal distinctive models we assign only the exclusive parts of CoreList and PhonList as training sets CoreTrain (18.025) and PhonTrain (950.955).

The common set EqualList can be further sub-divided into a subset EqualPron (30.724) with identical pronunciation encoding in CoreList and PhonList and a subset DiffPron (10.797) where the two encodings deviated at least by Levenshtein of 1. The entries from subset EqualPron are added to the training set CoreTrain (48.749 in total) of the valid pronunciations since the two training sets are un-balanced in favor to PhonTrain (950.955). The set DiffPron yields us a paired list of 10.797 encodings from which we can assume that the pronunciation stemming from PhonList is faulty while the pronunciation stemming from CoreList is valid.

For evaluation and parameter tuning (development set) we randomly extract 4 mutually-exclusive lists EvalList1-4 from the paired DiffPron list (2.699 pairs each) and use these in a leave-one-out schema for testing and development.

Figure 1 illustrates the partitioning of the used pronunciation lists.

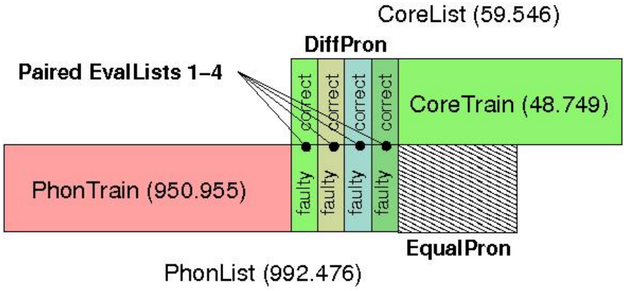

Figure 1: The partitioning of the two PHONOLEX pronunciation lists PhonList and CoreList into two training sets PhonTrain (list with errors) and CoreTrain (correct), and the 4 paired evaluations lists EvalList1-4 used for development and testing (see text for details).

2.3. Detection by Discriminant Model Comparison

By using standard counting techniques we train two different tri-gram models M_c and M_f on the training sets CoreTrain (*correct transcriptions*) and PhonTrain (*potentially faulty transcriptions*) respectively. To evaluate an unknown encoding A_W of a word W with regard to potential transcription errors we then calculate the difference of the probability estimates on both models:

$$D(A_W) = L(A_W, M_f) - L(A_W, M_c)$$

If A_W contains a phonotactically faulty encoding, $D(A_W)$ should tend to positive values, because model M_f should yield a higher probability estimate than M_c. If on the other hand A_W contains a correct encoding, we expect $D(A_W)$ to tend to negative values. Since L is logarithmic, the difference D is technically a normalization of probabilities; this normalization is necessary because the tri-gram probabilities in general (i.e. independent of possible encoding errors) are heavily dependent on the encoding A.

To test the feasibility of this concept we first calculated the D values for the total 4 EvalLists (10.797 words) where both, the correct encoding as well as a realistic faulty encoding, are known. Figure 2 shows the histograms of D for the correct (yellow) and faulty (blue) encoding. Both distributions overlap, but there is a tendency that the probability differences D of correct encodings concentrate on lower values than those of their faulty counterparts. A simple t-test shows that the distributions of D actually differs significantly between correct and faulty encoding ($p < 0.0001$).

Theoretically the optimal boundary between the two cases should be $D_b = 0$ but since the models M_c and M_f are trained to limited and unbalanced data sets, we have to calculate the optimal decision threshold from a development set using the Bayes criterion[7]. We estimate a Gaussian distribution on both histograms yielding two means m_1, m_2 and two standard deviations σ_1, σ_2.

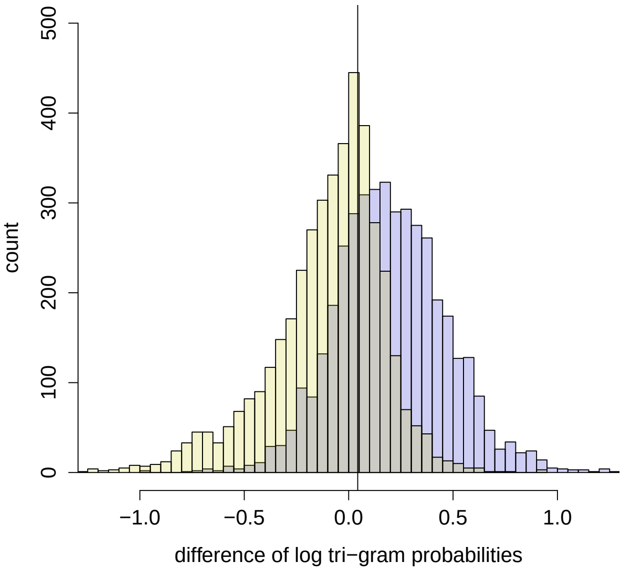

Figure 2: Histograms of tri-gram probability differences on EvalList1-4 for correct (yellow) and faulty (blue) encodings. The vertical black line denotes the optimal Bayes decision boundary based on the total evaluation set.

Then we analytically solve the equality of these Gaussian functions using the formula:[8]

$$\begin{aligned} D_{b1,2} &= \frac{1}{\sigma_1^2 - \sigma_2^2} \left\{ m_2\sigma_1^2 - m_1\sigma_2^2 \pm \right. \\ &\pm \left. \sigma_1\sigma_2\sqrt{(m_1 - m_2)^2 + 2(\sigma_2^2 - \sigma_1^2)\log\frac{\sigma_2}{\sigma_1}} \right\} \end{aligned}$$

where $D_{b1,2}$ are the points in the distribution where both Gaussians intersect. Finally we select the D_b which is positioned between the two means m_1, m_2.

In our case the empirical optimal Bayes boundary $D_b = 0.04$ (with regard to the total evaluation set EvalList1-4).

3. Application to Pronunciation Encoding

When producing a new pronunciation dictionary for a resource we recommend the following steps (see process flow diagram in Figure 3):

1. For the language in question prepare a database and estimate tri-gram models M_c and M_f and the optimal Bayes threshold D_b as described in Section 2..
2. Starting with the list of orthographic entries for which a pronunciation coding is needed (orth. input), first filter out orthographic keys for which a reliable source already provides a matching pronunciation coding and

[7] = identity of the (uni-modal) probability density functions

[8] Please note that this formula is only valid for distributions based on the same number of samples. For non-balanced data sets the sample counts N_1, N_2 of the histograms have to be taken into account in the log term: $log\frac{\sigma_2 N_1}{\sigma_1 N_2}$

pass them to the output un-inspected.
E.g. for German we recommend to use the Core part of PHONOLEX.

3. For the remaining orthographic entries run a Text-to-Phoneme algorithm to calculate a first version of transcripts.
   E.g. for German we recommend the BALLOON tool[9] (Reichel and Schiel, 2005).

4. Inspect the trigrams of each newly created transcription A: if at least one trigram of A has never been observed in your training sets, flag the the transcription for manual inspection.

5. Test each of the remaining transcriptions A by calculating $D(A)$ and compare this value to the optimal threshold D_b. If $D(A) > D_b$, flag A for manual inspection. Otherwise, pass A to the output un-inspected.

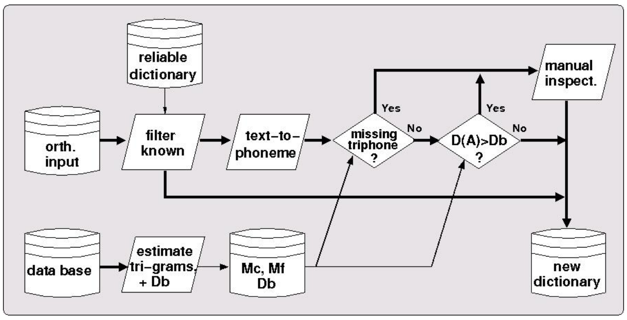

Figure 3: Process flow for the efficient creation of a new pronunciation dictionary.

4. Evaluation

To evaluate the automatic flagging of uncertain pronunciation encodings we simulate only the process steps 4-5 of the previous section. We use the 4 paired evaluation lists EvalList1-4 as described in Section 2.2. in a leave-one-out schema where 3 of the EvalLists are used to calculate the optimal threshold D_b (development set) and the remaining EvalList is used as the test set, i.e. the output of the (fictitious) TTP algorithm. Since the EvalLists are balanced, this means the assumed TTP produces exactly 50% pronunciation encoding errors. Correct/false acceptance/rejection rates are then averaged over the 4 tests. Table 1 shows the results. Columns add up to 50% each, since the test sets were balanced for correct and faulty encodings. In total 65.1% of encodings are flagged for manual inspection, 43.3% of these are truly faulty, while the remaining 21.8% are in fact correct. Precision[10] is 80.8% which is encouraging. On the other hand recall[11] is with 56.4% rather low, but this only means that 21.8% of the encodings have to be checked manually, while actually being correct. The worst case, a faulty encoding being passed unchecked (false positive), is with 6.7% rather low. On the other hand the actual reduction in effort (proportion of transcriptions not flagged for manual inspection) is 34.9%[12].

	correct encoding	faulty encoding
accepted	28.2%	6.7%
rejected	21.8%	43.3%

Table 1: Acceptance/rejection rates from the German evaluation on PHONOLEX.

In the evaluation we used a balanced set to achieve reliable estimates for true negative and false negative rates (lower line in Table 1). That means that our (fictitious) TTP algorithm produced 50% errors when calculating the new transcriptions. But for a real application we expect the error rate of the TTP algorithm to be much lower (somewhere in the range of 5-20%). Since our test sets were taken from a real application, we expect that the rate distribution for faulty encodings (right column) shown in Table 1 will hold for an un-balanced data set as well. Insofar the proportion 6.7% of faulty encodings in the output in our evaluation is too pessimistic. If we estimate the error rate of the TTP algorithm to be 20%, we expect only 3.1% of truly faulty encodings in the output.

We also looked more closely at the two error groups in Table 1:

- Correct encodings being rejected:
  About half of encodings in this group that were flagged because one or more triphones were unknown (step 4 in the process of Section 3.) are proper names or words of non-German origin, e.g. ’Utrecht, Bonbon, Croissant’.
  Words which were flagged because of the tri-gram models seem to have more than average syllable numbers, often in compound words, e.g. ’atmosphärischen, Asylfrage, Predigtverbot, Tschechoslowakei’.

- Faulty encodings being accepted:
  This group consists of

  – words with a wrong but phonotactically plausible encoding, e.g. ’womögliche’ encoded as /vo:m2:klIC@/ instead of /vOm2:klIC@/,

  – words of non-German origin that carry a ’Germanized’ pronunciation encoding, such as ’Siena’encoded as /zi:na/ instead of /zie:na/ oder ’Lunch’ encoded as /lUnC/ instead of /lanS/,

  – some proper names, and

  – some very few words that seem not to be German words but rather parts of words as occur frequently in conversational speech, e.g. ’egen, Tu, hare, öf’.

[9] A webservice WebBALLOON will be made available within the CLARIN-D project for public use.

[10] precision = true positive / (true positive + false positive)

[11] recall = true positive / (true positive + false negative)

[12] 28.2% + 6.7% = 34.9%

5. Conclusion

We studied the possibility to automatically flag newly created pronunciation encodings for manual inspection to reduce time and effort within the production of high-quality technical pronunciation dictionaries. Using normalized trigram model probabilities and a Bayes-optimized decision criterion we were able to achieve a precision of 80.8% on our test sets. 34.9% less entries were passed to the manual inspection (reduction in effort). Acceptance of unchecked but nevertheless faulty encodings was only 6.7% (false positives) on a balanced test set. For un-balanced test sets as expected in a real application of the method the proportion of false positives will be even lower and – where necessary – can be further decreased by skewing the decision threshold to values below the Bayes optimum. We conclude that the proposed method will reduce the overall effort and might be worth applying in future pronunciation dictionary productions.

6. Acknowledgments

This study was partly funded by the Bavarian Archive of Speech Signals (BAS). The authors thank Uwe Reichel of the CLARIN-D consortium for his valuable advice and fruitful discussions.

7. References

J. Cohen. 1960. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20(1):37–46.

The International Phonetic Association. 1999. The Handbook of the International Phonetic Association. Cambridge University Press:Cambridge, UK.

D. Jurafsky, J.H. Martin. 2009. Speech and Language Processing. Upper Saddle River, New Jersey, USA:Prentice Hall, 2nd edition, chapter 7.

M. Libossek, F. Schiel. 2000. In: *Proc. of the International Conference on Spoken Language Processing*, pages 283–286.

U.D. Reichel, F. Schiel. 2005. Using Morphology and Phoneme History to improve Grapheme-to-Phoneme Conversion. In: *Proceedings of the Interspeech 2005*, pages 1937-1940, Lisbon, Portugal:ESCA.

F. Schiel, Chr. Draxler, Ph. Hoole, H.G. Tillmann. 1999. New Resources at BAS: Acoustic, Multimodal, Linguistic. In: *Proceedings of the Eurospeech*, pages 2271–2274, ESCA:Budapest, Hungary.

J.C. Wells. 1997. SAMPA computer readable phonetic alphabet. In: Gibbon, D., Moore, R. and Winski, R. (eds.): *Handbook of Standards and Resources for Spoken Language Systems*, Part IV, section B, Berlin and New York: Mouton de Gruyter.