A STATISTICAL MODEL FOR PREDICTING PRONUNCIATION

Florian Schiel

Bavarian Archive of Speech Signals, Munich
schiel@bas.uni-muenchen.de

ABSTRACT

A general statistical model for the prediction of pronunciation given the orthographic transcript or the canonical pronunciation of a spoken utterance is described. The model is based on a Markov process that can be derived from a set of statistically weighted re-write rules. The automatic learning of such re-write rules based on annotated speech data is illustrated.

One possible application of the pronunciation model is the automatic phonetic segmentation and labelling of speech by augmenting the Markov process with Hidden Markov Models for phonetic segments. A publicly accessible system using the model for the automatic phonetic segmentation and labelling of 14 different languages is presented.

Keywords: pronunciation model, statistics, data-driven, phonetic segmentation, automatic segmentation

1. INTRODUCTION

Since most languages show intrinsic variation of pronunciation across their speech community depending on multiple factors such as speaker, speaker state, dialect, communication context etc., the prediction of realistic phonetic pronunciations given only the text form of a spoken utterance is non-trivial. A general pronunciation model for that purpose should accomplish two tasks:

1. Given a language (or dialect etc.) and a well-formed text within this language, generate all possible phonetic realisations Ψ in the form of strings of phonetic symbols K (e.g. encoded in IPA or SAM-PA)
2. For each generated $K \in \Psi$ calculate a-priori probabilities $P(K)$ that the particular phonetic realization K appears in spoken language.

In this contribution we propose a statistical pronunciation model that can be either based on a hand-crafted phonological model or machine-learned on annotated speech data. The model is independent of a lexicon and works across word boundaries.

Such a model is useful for a number of areas, such *as speech production modelling, dialect modelling/recognition, lexical modelling for automatic speech recognition/speech synthesis, automatic detection of phonological processes (e.g. [9]) or text-to-phoneme conversion.*

To demonstrate the usability of the proposed pronunciation model we will describe it in the context of *automatic phonetic segmentation and labelling* (S&L), and show how the model can be trained on annotated speech data.

2. PHONETIC SEGMENTATION AND LABELLING

Phonetically segmented and labeled speech corpora are required for many phonetic analyses and speech processing tasks. Manual segmentations are precise but inconsistent, since they are often produced by more than one labeler, and require time and money. Automatic S&L systems generate re-producible results, are much faster (often realtime), but not as precise as human labelers. Nevertheless project requirements often dictate the use of automatic methods. Practical applications of automatic S&L are nowadays in most cases implemented as a statistical search for a S&L $\hat{K}$ in a set Ψ of all possible S&Ls, which can be formulated as:[1]

(1)
$$\hat{K} = \text{argmax}_{K\in\Psi} P(K|O) = \text{argmax}_{K\in\Psi} \frac{P(K)p(O|K)}{p(O)}$$

where O is the corresponding speech signal. Since the probability for the speech signal $p(O)$ is a constant for all K this can be reduced to the simple well known formula

(2) $\hat{K} = \text{argmax}_{K\in\Psi} P(K)p(O|K)$

where $p(O|K)$ models the probability (density) of the acoustics given a certain (discrete) S&L (e.g. by using HMM, ANN etc.) while $P(K)$ models the apriori probability of the symbol sequence in the S&L K ([3]).

Automatic S&L systems mainly differ in the nature of the search space Ψ and the way that $P(K)$ is modeled, i.e. is calculated for a given $K \in \Psi$. For example, a simple *forced alignment* to a given phonemic transcript $\hat{K}$ yields

(3) $||\Psi|| = 1$ and $P(\hat{K}) = 1$

since there is only one possible S&L and hence only $p(O|K)$ is maximized here by searching for the alignment with the highest probability.

It has been shown for several languages that more sophisticated S&L approaches which successfully predict a realistic search space Ψ yield better S&L results than a simple forced alignment to a single most likely or even canonical pronunciation (see for example [1, 8, 10]).

The calculation of Ψ and $P(K)$ does not necessarily require a statistical model. For instance the SPPAS system described in [1] first tokenizes the spoken text into words and then performs a lookup in a vast pronunciation dictionary with multiple entries for each word token. The identified pronunciations are then chained group-wise after one another so that each variant has equal probability within a group, i.e. all possible combinations of pronunciation variants along the chain have the same probability. The acoustical model then tracks the most likely combination of variants by matching the complete chain to the speech signal.

In [10] and [4] Ψ is determined by applying phonological pronunciation rules to a canonical pronunciation of an utterance yielding M alternative pronunciation variants which are then treated with the same probability $P(K) = \frac{1}{M}$ in the search.

Other ways to model $\Psi, P(K)$ include using an n-gram phonotactic model, a lexicon of pronunciation variants with conditional probabilities or a Markov process on words/syllables/phonemes.

In the S&L approach exemplified in this paper Ψ is modelled for a spoken text of arbitrary length by building a Markov process $\mathscr{G}(N,A)$ with phonemic symbols in the nodes N and transition probabilities on the arcs A. Each path from the start node to the end node represents a possible $K \in \Psi$ and accumulates to the probability $P(K)$. $p(O|K)$ is determined by HMMs for each phonemic segment and a simple Viterbi search through the graph yields the maximal $P(K)p(O|K)$ and by backtracking the path through $\mathscr{G}$ $\hat{K}$ is determined; this technique is used in the Munich AUtomatic Segmentation (MAUS) system ([8]).

This paper describes the method to build an automaton for the structural core of the pronunciation model $\mathscr{G}$ (Section 3), the machine-learning of statistically weighted pronunciation re-write rules from an annotated speech corpus (Section 4), and the conversion of the basic automaton into a Markov process for S&L (Section 5). Finally, Section 6 gives some practical hints for the usage of the implemented S&L system MAUS.

3. BUILDING THE AUTOMATON

Input to the process is a string of orthographic words representing the spoken utterance. The orthographic form is transformed into a linear citation pronunciation form, called the *canonical form* $\mathscr{C}$ hereafter. This can be done either by lexicon lookup or a text-to-phoneme system, or – as in the case of MAUS – a combination of both. The canonical form $\mathscr{C}$ can be represented by a simple left-to-right finite-state automaton $\mathscr{G}_c(N,A)$ without self transitions where each node emits exactly one phonemic symbol; the first and last states are non-emitting enter and exit states.

$\mathscr{G}_c$ can now be extended by additional arcs, emitting and non-emitting states to model variations from the canonical form. Technically this is done by applying a set of matching re-write rules where each rule is defined by a tuple (a,b,l,r) with a pattern string a, a replacement string b and left/right context strings l,r. Essentially each application of a rule creates a new arc with a number of new nodes (or zero). b,r,l may also be the empty string $\emptyset$ to allow for deletions of symbol strings as well as non-defined contexts; insertions are modelled by a replacement string b that is an extension of the (non-empty) pattern string a. In addition the symbol /#/ may be used to model word boundaries, to allow the modeling of cross-word effects or word initial/final contexts. Since re-write rules are only applicable to the canonical form (the sub-automaton $\mathscr{G}_c$) a single pass over the rule set creates an automaton $\mathscr{G}$ covering all possible pronunciation variants (with no recursive applications of rules required); the problem of over-lapping empty contexts $l,r = \emptyset$ can be solved by inserting non-emitting nodes (see [3], pp. 75-81).

Consider for instance the canonical form of the German word *Abend* (‘evening’)[2]:

/? a: b @ n t/

To model the very common reduction/assimilation processes that lead to the realizations

/? a: b m t/ and /? a: m t/

we need two re-write rules:

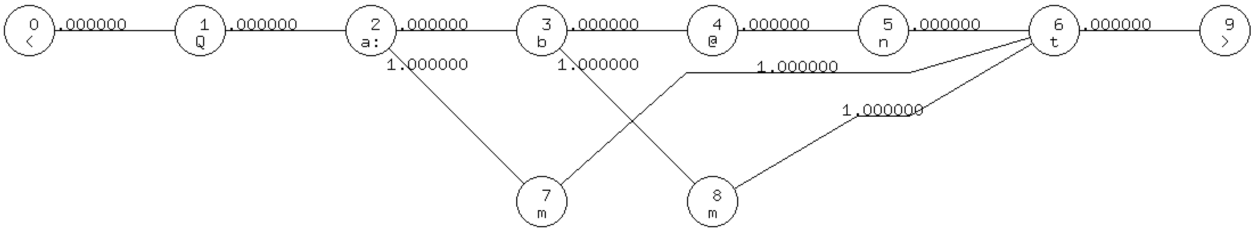
(/@n/,/m/,/b/,/t/) (/b@n/,/m/,/a:/,/t/)

resulting in the automaton shown in Figure 1.

4. CREATING THE RULE SET

In principle the set of re-write rules can be derived from different sources: they can be either data-driven (see below) or hand-crafted. In the latter case the rule set often represents a phonological model

Figure 1: *Example Markov process for the word 'Abend'. /</ and />/ are non-emitting states.*



of the language concerned and contains no information about the probabilities of its possible substitutions. Although this most likely deteriorates the performance of the S&L we sometimes use this technique for languages where there is not enough annotated training material available or when dealing with special speech recordings documenting well-known phonological processes as is quite common in phonetic studies.

The pronunciation model proposed here achieves the best performance when using a data-driven statistical weighted rule set. Rules (a,b,l,r) can be found by performing a *longest common subsequent alignment* ([3]) between the canonical form $\mathscr{C}$ and the annotation (the realization) $\mathscr{R}$ of a recorded utterance and then segmenting the alignment according to common and deviating portions. When restricting the left/right context l,r of each rule to length 1, it is quite straightforward to extract rules (a,b,l,r) from each deviating portion of the alignment and determine their total number $n(a,b,l,r)$ from the annotated corpus. In parallel, the total number of occurrences of the string (l,a,r) can be derived from the canonical forms $\mathscr{C}$ of the corpus: $n(l,a,r)$. Using maximum-likelihood we can then estimate the conditional probability of the application of a rule by:

$$(4) \quad \hat{P}(b|l,a,r) = \frac{n(a,b,l,r)}{n(l,a,r)}$$

Since manually annotated speech corpora are rare and in most cases small, simple maximum-likelihood estimates may not generalize sufficiently. There are two possible ways to yield a more robust rule set:

1. Use a discounting technique to spread probability mass to all unseen rule contexts (l,a,r). This leads to an explosion of the rule set and subsequent computational problems. Therefore it is necessary to prune the rules, for instance according to phonotactics.
2. Vary the context of each rule with non-empty left and right context into unseen but plausible new rules. The basic idea here is that since left and right context might be statistically independent, the system might encounter pronunciation variants with only either the left or the right context or new combinations of those.

In our S&L system we use the second approach, since it proved to be more robust than the discounting technique but we restrict the context spreading to phonetically similar classes (e.g. a context /n/ is spread to contexts /n m N/ only).

5. FROM AUTOMATON TO MARKOV MODEL

Up to this point we have created a finite-state automaton that covers all hypothetical realizations predicted by the rule set. To use this automaton effectively for a combined acoustical/phonotactic Viterbi search, we need to augment it with probabilities for emissions and transitions, thus creating a true Markov process. This is not a trivial task since the automaton may model paths (= phoneme sequences) of different length, but still every path K through the model must yield the appropriate accumulated probability $P(K)$.

For the intended purpose of phonetic S&L we can define the *emission probability* of each node as the production probability of a Hidden Markov Model (HMM) that is trained to manually segmented training samples of the corresponding label classes. In other words we replace the nodes N of $\mathscr{G}(N,A)$ by HMMs for label classes. Other applications of the pronunciation model (as mentioned in the Introduction) might require a different approach here depending on the task.

Regarding the generally required *transition probabilities* between nodes we may distinguish two main cases:

1. Accumulated probabilities are assumed to be equally distributed over all hypothesized realizations $\mathscr{R} \in \Psi$. For instance, if no statistical information about rule application is available.
2. Accumulated probabilities must reflect the

$\hat{P}(b|l,a,r)$ of all applied re-write rules (a,b,l,r) (as defined in Section 4) along a path $\mathscr{R}$ through the model.

Due to space constraints we will only demonstrate the recursive calculation of the correct transition probabilities using the (easier) first case, and will also not consider the additional problem of different rules with overlapping contexts (see [3], pp. 98-102 for details on the second case).

We define the *rank* of a node d_i as its distance from the non-emitting starting node (the starting node has rank 0), the set $\Gamma^-(d_i)$ as the set of all nodes that precede a node d_i and $\Gamma^+(d_i)$ as the set of nodes that follow d_i. Let $N(d_i)$ be the number of possible paths that end in node d_i, which equals the sum of all paths ending in preceding nodes of d_j. $N(d_i)$ can therefore be calculated for all nodes in ascending rank order by applying the recursive formula

(5)

$$N(d_j) = \begin{cases} 1 & \text{for the starting node} \\ \sum_{d_i \in \Gamma^-(d_j)} N(d_i) & \text{else} \end{cases}$$

$P(d_i)$, the probability that a node is part of a phoneme sequence can also be calculated for all nodes, since we know that this probability must be 1 for the last node, and we can recursively calculate the probabilities with descending rank order using

(6)

$$P(d_i) = \sum_{d_j \in \Gamma^+(d_i)} P(d_j)P(d_i|d_j) = \sum_{d_j \in \Gamma^+(d_i)} P(d_j)\frac{N(d_i)}{N(d_j)}$$

Since the model is acyclic and we consider all paths through the model as equally probable, we can say that the backward probability that a node d_i precedes a node d_j is

$$(7) \quad P(d_i|d_j) = \frac{N(d_i)}{N(d_j)} \quad \text{with} \quad d_i \in \Gamma^-(d_j)$$

By applying Bayes we get the desired transition probability from node d_i to node d_j:

$$(8) \quad P(d_j|d_i) = \frac{P(d_j)N(d_i)}{P(d_i)N(d_j)} \quad \text{with} \quad d_i \in \Gamma^-(d_j)$$

which can then be calculated for each transition found in the model. The result of this procedure is then a true Markov process with integrated HMMs to calculate the emission probabilities. It can be used in standard software for automatic speech recognition (e.g. the HTK package[3]) to find the sequence $\hat{K}$ together with the segmentation of the aligned speech signal with the highest combined probability. Thus in one decoding step both the phonetic transcript and the segmentation can be obtained.

6. MAUS SOFTWARE PACKAGE AND WEBSERVICE

The MAUS system is available as freeware at the Bavarian Archive of Speech Signals (BAS, [6]). The current version 2.102 covers the languages *American English, Australian English, Dutch, English, Estonian, Finnish, Georgian, German, Hungarian, Italian, New Zealand English, Polish, Portuguese, and Spanish.* MAUS also features a 'language independent' mode with pure SAM-PA input to process yet unsupported languages or languages that have no orthographic writing system.

The core tool *maus* performs an automatic S&L on a single recording, starting with the canonical phonemic transcript. In combination with a text-to-phoneme tool (e.g. BALLOON ([7]) it can provide a phonetic S&L based only on the orthographic transcription and the speech signal. Other tools within the freeware package allow the segmentation of a whole speech corpus or iterative segmentation with adaptation of the acoustical model to the target speech data. Output can be produced in SAM-PA, IPA, manner or place of articulation.

To simplify the usage several webservices are available on the BAS CLARIN server ([5, 2]). The webservices can be accessed via a web interface[4] or by RESTfull calls enbedded in user applications[5].

7. DISCUSSION

The proposed model has the advantage that it provides comparable probabilities for different pronunciation variants, even if these have different lengths (which is usually the case). The statistical model works independently of the nature of the underlying re-write rules, although the best results so far have been obtained with machine-learned rules derived from a large annotated speech corpus. The method is applicable for very large rule sets (we have tested with rule sets larger than 10,000 rules), because it exploits partial overlapping sequences effectively; simply generating all possible pronunciation variants from a large set of rules often causes a model to explode exponentially. We demonstrated the usefulness of the model in the context of phonetic segmentation and labelling, but in principle the same model can be applied to other cases that require a statistically sound prediction of phonetic pronunciation.

8. REFERENCES

[1] Bigi, B. 2012. SPPAS: a tool for the phonetic segmentations of speech. *Proc. of the LREC 2012* Istanbul, Turkey. 1748–1755.

[2] Common Language Resources and Technology Infrastructure (CLARIN). http://www.clarin.eu/.

[3] Kipp, A. 1998. *Automatische Segmentierung und Etikettierung von Spontansprache*. Munich, Germany: Doctoral Thesis, Ludwig-Maximilians-Universität.

[4] Kipp, A., Wesenick, M., Schiel, F. 1996. Automatic Detection and Segmentation of Pronunciation Variants in German Speech Corpora. *Proc. of the ICSLP* Philadelphia, USA. 106–109.

[5] Kisler, T., Schiel, F., Sloetjes, H. 2012. Signal processing via web services: the use case webmaus. *Proceedings of the Digital Humanities 2012* Hamburg, Germany. 30–34.

[6] Freeware package 'Munich AUtomatic Segmentation' (MAUS). ftp://ftp.bas.uni-muenchen.de/pub/BAS/SOFTW/MAUS.

[7] Reichel, U. 2012. PermA and Balloon: Tools for string alignment and text processing. *Proc. of the Interspeech 2012* Portland, Oregon. paper no. 346.

[8] Schiel, F. 1999. Automatic phonetic transcription of non-prompted speech. *Proc. of the ICPhS* San Francisco, USA. 607–610.

[9] Schiel, F., Stevens, M., Reichel, U. D., Cutugno, F. 2013. Machine Learning of Probabilistic Phonological Pronunciation Rules from the Italian CLIPS Corpus. *Proceedings of the Interspeech 2013* Lyon, France. 1414–1418.

[10] Wester, M., Kessens, J., Strik, H. 1998. Improving the performance of a Dutch CSR by modeling pronunciation variation. *Proc. of the Workshop on Modeling Pronunciation Variation* Rolduc, Netherlands. 145–150.

[1] without loss of generality we include the segmental information in the pronunciation model Ψ and K hereafter.

[2] phonemic symbols in SAM-PA;
www.phon.ucl.ac.uk/home/sampa/;
last accessed 2015-01-22

[3] htk.eng.cam.ac.uk; last accessed 2015-01-22

[4] clarin.phonetik.uni-muenchen.de/BASWebServices/,
last accessed 2015-01-22

[5] see clarin.phonetik.uni-muenchen.de/
BASWebServices/services/help,
last accessed 2015-01-22, for getting started with REST based usage