Issues in Building General Letter to Sound Rules

Alan W Black[1] *Kevin Lenzo*[2] *Vincent Pagel*[3]

[1]awb@cstr.ed.ac.uk, CSTR, University of Edinburgh,
[2]lenzo@cs.cmu.edu, Carnegie Mellon University,
[3]pagel@tcts.fpms.ac.be, Faculté Polytechnique de Mons

ABSTRACT

In general text-to-speech systems, it is not possible to guarantee that a lexicon will contain all words found in a text, therefore some system for predicting pronunciation from the word itself is necessary.

Here we present a general framework for building letter to sound (LTS) rules from a word list in a language. The technique can be fully automatic, though a small amount of hand seeding can give better results. We have applied this technique to English (UK and US), French and German. The generated models achieve, 75%, 58%, 93% and 89%, respectively, words correct for held out data from the word lists.

To test our models on more typical data we also analyzed general text, to find which words do not appear in our lexicon. These unknown words were used as a more realistic test corpus for our models. We also discuss the distribution and type of such unknown words.

1. INTRODUCTION

Given that lexicons are closed by their nature and that the input text for general text to speech (TTS) systems is open, there will always be words in the text which are not contained within even the largest lexicon. Even when a large lexicon can be constructed to cover the whole vocabulary it would be useful to find a principled method to reduce the size of the lexicon (which we discuss more fully in [11]).

In many languages the orthographic system has some relationship to the pronunciation, depending on the language it may be trivial (such as in languages like Spanish) or relatively difficult (like English), or harder (such as in Japanese with full kanji). Humans can (often) pronounce words reasonably even when they have never seen them before. It is that ability we wish to capture automatically in an LTS rule system.

Here we present a method for taking large lists of words and pronunciations and building generalized rule systems that not only produce reasonable pronunciations for unseen words but also allow us to remove the regular examples from the list so that much smaller lexicons are adequate for the same coverage.

2. LETTER-PHONE ALIGNMENT

In order to make the building of models easier we wish to have a standardized alignment between the letters in an entry and the phones in its pronunciation.

The number of letters in a word and the number of phones in its pronunciation in general are not a one to one match. For the languages we have investigated, letters can map to zero, one, two or very exceptionally three phones. Even when there are the same number of letters and phones the "correct" alignment may not be the most simple. In general there seems to be less phones than letters.

The cases where a letter goes to more than one phone are fairly restricted (e.g. `x` to `/k s/`, `o` to `/w uh/` as in `one`). Almost all letters can in some context correspond to no phone, which we will call `_epsilon_`.

A more complex model involving multi-letter clusters to zero or more phones is also possible though this introduces complexities in the model learning, and alignment process that we preferred to avoid.

Ideally we would like a purely automatic method for finding the best single letter alignments, but so far we have achieved better results from a hand-seeded method.

The hand-seeded method requires the explicit listing of which phones (or multi-phones) each letter in the alphabet may correspond to, irrespective of context. This is relatively easy to do and can be done as an interactive process over the training set as new correspondences are added to the allowables list. For example the letter "c" may be realized as any one of

epsilon k ch s sh t-s

Vowel letters have typically a much longer list of potential phones.

The hand-seeded algorithm takes the list of allowables and finds all possible alignments between each entry's letters and phones. A count is taken for which correspondences are used for each alignment and a table of probabilities of a phone (epsilon or multi-phone) given a letter is estimated, again irrespective of context. Then the entries are re-aligned and each possible alignment is scored with the generated probabilities. The best alignment is selected. The alignments generated by this algorithm are close to what would be produced by hand and it is very rare to find alignments that would be considered unacceptable.

The building of the allowables table is simple and quick though does require some skill, however it can be done even without an in-depth knowledge of the language the lexicon is for. A few words do not produce alignments (which would require new entries in the allowables table) which typically represent classes for which the relationship between the letter form and the phones is too opaque. These are typically abbreviations, such us "dept" as /d ih p aa r t m ah n t/; words with very unusual pronunciation e.g. "lieutenant" (British English); Foreign words (e.g. "Lvov") and what could be considered mistakes in the lexicon e.g. "cannibalistic" with two /l/ phones. Typically the number of entries that failed to have an alignment are well under 1%.

The second alignment is an application of the expectation maximization (EM) algorithm [7] which we call the "epsilon scattering method". The idea is to estimate the probabilities for one letter L to match with one phoneme P, and to use DTW to introduce epsilons at positions maximizing the probability of the word's alignment path. Once the dictionary is aligned, the association probabilities can be computed again, and so on until convergence. e.g. five iterations are necessary on the CMU lexicon

Algorithm:

```
/*  initialize prob(L,P) */
1 foreach word in training_set
     count with DTW all possible L/P
     association for all possible epsilon
     positions in the phonetic
     transcription
/* EM loop */
2 foreach word in training_set
     compute new_p(L,P) on alignment_path
3  if (prob != new_p) goto 2
```

This differs from [6] in that the probabilities are distributed equally ('scattered') among each of the possible alternatives, rather than assigning an arbitrary weight to each shift.

When we build models from the results of alignment using each of the above algorithms on the OALD we get the follow results

Method	Letters	Words
Epsilon scattering	90.69%	63.97%
Hand-seeded	93.97%	78.13%

"Letters correct" is the number of letter-phone pairs which are correctly predicted with respect to the test set. "Words correct" are the number of complete words where the complete phone string predicted (minus epsilons, but including stress markers) is correct with respect to the test set.

So we can see clearly that the hand-seeded method is better. However we still feel that the hand-seeded is a simple task and feel that we have not yet fully investigated method to improve the automatic method to achieve the level of the hand-seeded method.

3. BUILDING RULES

Once an alignment is found we can train a phone prediction model. In our work we have used decision tree technology [3] as we feel this is simple and produces compact models. We also feel that other learning techniques would not produce significantly better results.

For each letter in the alphabet of the language we trained a CART tree given the letter context (three either side) to predict epsilon, phone or double phone from the aligned data. One can build a single tree without any significant difference in the accuracy but building separate trees is faster and allows for parallelization.

We split the data into train and test data by removing every tenth word from the lexicon. This means that the data set contains only one occurrence of each word and hence word frequency is ignored. Another factor is that as these lexicons usually contain many morphological variations, it is likely there will be a similar word or words in the training set.

We removed short words (under four letters) from the training and test sets as these words are typically function words which in general may have non-standard pronunciations, or are abbreviations (e.g. "aaa" as /t r ih p ah l ey/) which have little or no relationship with their pronunciation. Also, where part of speech information was available, we removed all non-content words. The reasoning is that unknown words are typically not the most common words and in general unknown words will have more standard pronunciations rather than idiosyncratic ones.

We have so far tried this technique on four lexicons, Oxford Advanced Learners Dictionary of Contemporary English (OALD) (British English) [10], CMUDICT (US English) [4], BRULEX (French) [5] and the German Celex Lexicon [1].

	Correct	
Lexicon	Letters	Words
OALD	95.80%	74.56%
CMUDICT	91.99%	57.80%
BRULEX	99.00%	93.03%
DE-CELEX	98.79%	89.38%

CMUDICT, although also English, does not get as good results compared with OALD as it contains many more "foreign" words, particularly names, which are much harder to predict without any higher level information (such as ethnic origin).

The above results are the best results achieved after testing various parameters in the CART building process. Particularly we varied the "stop" value which specifies the minimum number of examples necessary in the training set before a question is hypothesized to distinguish the group. Normally the smaller the stop value the more over-trained the models may become. However the following table shows the results for OALD, tested on held out data, while changing the stop value

Stop	Correct		Size
	Letters	Words	
8	92.89%	59.63%	9884
6	93.41%	61.65%	12782
5	93.70%	63.15%	14968
4	94.06%	65.17%	17948
3	94.36%	67.19%	22912
2	94.86%	69.36%	30368
1	95.80%	74.56%	39500

As the stop value is reduced, the size of the model increases. The model size is the total number of questions and leaf nodes in the generated CART trees. However it appears that more finely tuned data is always better, such that even with stop value 1 the model is not over-trained.

Note that comparisons with other LTS training techniques are not that easy. As when the train/test sets differ, and when the domains differ there can be no direct comparisons. For example if we remove proper names from the OALD and train and test on the remainder our word correct score goes up to 80%. However the above results compare favorably with other systems using similar data sets (e.g. [8]).

4. STRESS ASSIGNMENT

The importance, and realization of lexical stress varies between languages but in order to produce a reasonable pronunciation from a string of letters it is often more than simply producing a string of phones, lexical stress markings are also required. In English lexical stress may be different depending on syntactic class, it may even move with some morphological derivations. Therefore predicting lexical stress for each vowel in the predicted string cannot in general be done from the letter context alone. However results in [12] suggest that combining phone and stress prediction in a single model give better results.

We tested this on the OALD data set. We first built letter to phone models where lexical stressing information was removed from the phones and we trained a separate stress prediction model using the same test set using features such as syllable position in word, vowel length, vowel height, number of syllables from end of word, and part of speech. On held out data from the OALD the per syllable results are

Actual	Predicted		
	unstressed	stressed	%
unstressed	7390	378	95.1%
stressed	512	8207	94.1%

total correct 15597/16487 (94.6%)

This model was combined with the output of the letter to phone model (LTP+S).

The second model introduced two types of vowel phone, stressed and unstressed versions. The standard LTS model building technique was applied so the CART trees themselves produced phone and stressing information directly (LTPS).

	LTP+S	LTPS
LNS	96.36%	96.27%
Letter	—	95.80%
WNS	76.92%	74.69%
Word	63.68%	74.56%

(LNS = letter/phone ignoring stress, WNS = word ignoring stress)

A score for “letters correct” for the separate model is not available as the stress prediction model does not preserve alignment.

Thus it can be clearly seen that although higher values are possible per word when ignoring stress, a separated model applied afterwards gives significantly lower results than if the phones and stress levels are predicted by a single model.

We also discovered that including part of speech information in the phone prediction models themselves improved the accuracy of the model. Without POS information the combined model gives 95.32% letter correct and 71.28% word correct. Thus part of speech obviously helps and is readily available in a TTS system with a standard POS tagger even for unknown words.

Ultimately stress cannot be predicted on local context alone as there are a number of example in English where local context is insufficient (cf. photograph/photography). Ideally morphological decomposition is required to do such prediction but we have not yet investigated this area.

5. DOES IT REALLY WORK

To find out a more realistic assessment of these models' treatment of unknown words we processed the first section of the WSJ Penn Treebank [9]. This consists of a total of 39923 words in news text style. Using our standard OALD lexicon we find that a total of 1775 words (4.6%) are not found in the lexicon, 943 of which are unique. Of those unknown words we find the following distribution

	Occurs	%
names	1360	76.6
unknown	351	19.8
American spelling	57	3.2
typos	7	0.4

American spelling of words is distinguished here (e.g. “honor”, “center”) as it is so systematic. As OALD is a British English Lexicon it doesn't contain such spellings, though for TTS use it obviously should. As WSJ is more carefully published than other texts such as email, the issue of typos is almost negligible. We have done similar analysis of unknown words from Time magazine articles finding a very similar distribution and ratio of unknowns, thus we feel the above is typical of news story type text.

We listened to each of the 1775 words as pronounced by a number of the models discussed above. A yes/no decision was made about acceptability. Note that a number these words have multiple acceptable pronunciations. If any of those were predicted they were deemed acceptable. For example the pronunciations of "Reagan" as /r ey g ah n/ and as /r iy g ah n/ were both considered acceptable.

The best results, shown above for OALD, were obtained by building the deepest possible trees. But when those models were applied to these unknown words the results showed that although the models were not over-trained for the unseen test set extracted from the lexicon itself, they were for these unknown words. The following shows the results after varying the stop value for CART building.

Stop	Lexicon Test set	Unknown Test set	size
1	74.56%	62.14%	39500
4	65.17%	67.66%	17948
5	63.15%	70.65%	14968
6	61.65%	67.49%	12782

Thus the best model for unknown words is not the best model for the held out lexical entries. What is more, the best model for unknown words is less than 40% the size of the best model for the lexical test set. These figures reflect both the fact that the held out data in the lexical test set (every tenth entry) is often just a morphological variation of the entries around it, and secondly the lexical test set does not take into account word frequency of unknown words.

Looking at those words that are pronounced wrongly we find some mistakes are still recognizable (e.g. Chrysler as /k r ih s l ah er/) but many are unacceptable and unrecognizable showing there is still work to be done. Further analysis of these words shows

	Occurs	%
names	413	79
unknown	94	18
American spelling	7	0
typos	2	0

One would expect proper names to be the hardest to pronounce (especially those of foreign origin) but although it appears they are slightly harder our model seems to do as well on them as other non-names.

Further analysis of the types of names that are still unpronounceable shows a larger proportion of non-anglo-saxon origin than in those that are correctly pronounced. As many of the languages these names originate from often have a more standardized pronunciation than English (e.g. Polish, Italian, Japanese (in its romanized form)), knowing the origin of an unknown word may allow more specific rules to be applied, but we have not yet investigated this area.

6. SUMMARY

We have presented automatic (and near automatic) processes for building letter to sound rules systems from lists of entries and their pronunciations. We have successfully built LTS models for four different languages and feel confident this process will work for many other languages. As well as quoting results from held out data from the words lists used for training, we also present results of applying one model to unknown words from news text.

This method is fully implemented and documented and distributed with the Festival Speech Synthesis System [2], or a PERL implementation is available from `http://www.cs.cmu.edu/ lenzo/t2p`.

ACKNOWLEDGEMENTS

We gratefully acknowledge the support of the UK Engineering and Physical Science Research Council (EPSRC grant GR/K54229), the US National Science Foundation graduate research fellowship scheme, and Oregon Graduate Institute for providing access to the German CELEX lexicon.

7. REFERENCES

1. R. Baayen, R. Piepenbrock, and L. Gulikers. The CELEX lexical database (cdrom). Linguistic Data Consortium, University of Pennsylvania, Philadelphia, 1995.
2. A. Black, P. Taylor, and R. Caley. The Festival speech synthesis system. http://www.cstr.ed.ac.uk/projects/festival.html, 1998.
3. L. Breiman, J. Friedman, R. Olshen, and C. Stone. *Classification and Regression Trees*. Wadsworth & Brooks, Pacific Grove, CA., 1984.
4. CMU. Carnegie Mellon Pronuncing Dictionary. http://www.speech.cs.cmu.edu/cgi-bin/cmudict, 1998.
5. A. Content, P. Mousty, and M. Radeau. Une base de données lexicales informatisée pour le francais écrit et parlé. *L'Année Psychologique*, 90:551–566, 1990.
6. W. Daelemans and A. van den Bosch. Language-independent data-oriented grapheme-to-phoneme conversion. In J. van Santen, R. Sproat, J. Olive, and J. Hirschberg, editors, *Progress in speech synthesis*, pages 77–90. Springer Verlag, 1996.
7. A. Dempster, N. Laird, and D. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*, 39 (Series B:1–38, 1977.
8. R. Luk and R. Damper. Stochastic phonographic transduction for english. *Computer Speech and Language*, 10:133–153, 1996.
9. M. Marcus, B. Santorini, and M. Marcinkiewicz. Building a large annotated corpus of English: the Penn Treebank. *Computational Linguistics*, 19:313–330, 1993.
10. R. Mitten. Computer-usable version of Oxford Advanced Learner's Dictionary of Current English. Oxford Text Archive, 1992.
11. V. Pagel, K. Lenzo, and A. Black. Letter to sound rules for accented lexicon compression. In *ICSLP98*, Sydney, Australia., 1998.
12. A. van den Bosch, T. Weijters, and W. Daelemans. Modularity in inductive-learned word pronunciation systems. In *Proc. NeMLaP3/CoNNL98*, pages 185–194, Sydney, 1998.