

Grapheme to Phoneme Conversion and Dictionary Verification Using Graphonemes

Paul Vozila, Jeff Adams, Yuliya Lobacheva, Ryan Thomas

Language Modeling Research, Dragon R&D
Scansoft, Inc. USA

paul.vozila, jeff.adams, yuliya.lobacheva, ryan.thomas@scansoft.com

Abstract

We present a novel data-driven language independent approach for grapheme to phoneme conversion, which achieves a phoneme error rate of 3.68% and a pronunciation error rate of 17.13% for English. We apply our stochastic model to the task of dictionary verification and conclude that it is able to detect spurious entries, which can then be examined and corrected by a human expert.

1. Introduction

Grapheme to phoneme conversion is an important feature in many speech systems. It is typically applied to words encountered by the system that do not appear in an *a priori* fixed lexicon. For example, in a voice dictation system like *Dragon NaturallySpeaking*, grapheme to phoneme conversion is required to provide pronunciations for new words added by a user.

Both manually constructed rules and data-driven techniques [1,2,3,4,5] have been applied to the grapheme to phoneme conversion task. The data-driven approaches include the use of finite state transducers [2], hidden Markov models (HMMs) [1], decision trees [1], latent semantic analysis [3], ngram models [4], and multigram models [5]. Data-driven approaches have the advantage of requiring no human expertise and typically offering language independence. These approaches begin with a training dictionary consisting of pairs of word spellings and pronunciations. A common starting point is to partition each entry into corresponding grapheme-phoneme sequence pairs (referred to as *graphonemes* in this paper) and to build a probabilistic model based on these units. Our procedure consists of an initial unit selection phase using HMMs, followed by a concatenative procedure for unit selection refinement, and finally an ngram model estimation stage based on these units.

A pronunciation dictionary is an integral component of any speech system. The dictionary is modified manually, potentially by multiple people, and hence is prone to typographical errors (in spellings and pronunciations) as well as consistency errors. Dictionary errors typically lead to system errors, as they are unlikely to be compensated for by other system components. Our graphoneme ngram model provides an estimate of the likelihood of a particular word spelling paired with a given phoneme sequence. Given a dictionary entry, we can mark it as suspicious if the likelihood of the dictionary pronunciation is sufficiently small relative to the most likely phoneme sequence as determined by the model. This allows us to flag suspicious dictionary entries and present them to a human expert sorted by our confidence that an entry is in error.

The structure of this paper is as follows. The graphoneme model training procedure is described in Section 2. We describe an efficient method for using the model to estimate the likelihood of a word spelling-pronunciation pair in Section 3. The procedure for dictionary verification is given in Section 4. In Section 5, we present a set of experiments using the graphoneme model. Finally, in Section 6 we conclude and discuss future work.

2. Graphoneme Model Training

Model training consists of three steps. Initial graphoneme unit selection is performed using HMMs and is described in Section 2.1. A concatenative unit refinement step, described in section 2.3, is then enacted using techniques formerly applied to word phrase language modeling. Finally a graphoneme ngram model is estimated as described in Section 2.2.

2.1. Initial Unit Selection

This phase begins with a training dictionary consisting of paired word spellings and pronunciations. For words with multiple pronunciations, each pronunciation represents a separate entry. For a given entry, we presume that the word's spelling was *generated* by the sequence of phonemes making up its pronunciation.

In particular, we assume that we first choose the number of graphemes that a given phoneme will produce (up to some maximum) and then choose the actual graphemes realized, with both decisions depending only on the phoneme identity [1]. An HMM as pictured in Figure 1 is used to describe these phoneme models. Concatenating appropriate phoneme models with epsilon transitions allows us to construct word-level models.

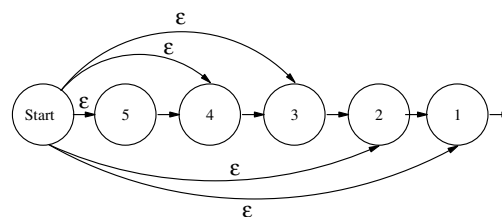


Figure 1. Phoneme HMM emitting up to 5 graphemes via numbered state transitions. The output distributions are tied.

Starting from uniform distributions, maximum likelihood phoneme model distributions are estimated via the Baum-Welch algorithm. Using these trained models, for each dictionary entry we compute the most likely state sequence that would have produced the given pronunciation via the

Viterbi algorithm. This results in each entry being segmented into a sequence of units where each unit consists of a single phoneme and zero or more graphemes. These units represent our initial set of graphonemes. A Viterbi segmentation for the word "thoughtfulness" and its pronunciation is given in Figure 2.



Figure 2. Viterbi segmentation of "thoughtfulness" and its pronunciation based on phoneme HMMs.

Note that this initial unit selection can also be done assuming that graphemes *generate* phonemes instead of the reverse. The current approach seems intuitively appropriate for English and indeed results in superior performance. However, for Japanese we have used the "graphemes generate phonemes" assumption with success.

2.2. Ngram Model Estimation

At this point each word spelling-pronunciation pair is represented by a sequence of graphonemes. We can complete each sequence using a special start-of-word symbol ("*<s>*") and end-of-word symbol ("*</s>*"). Based on these sequences a graphoneme ngram model is estimated using backoff smoothing and absolute discounting [9].

By summing over all complete sequences which produce the appropriate grapheme and phoneme strings, we can use this model to estimate the joint likelihood of a given word spelling-pronunciation pair. We can obtain the Viterbi approximation of this probability by considering only the most likely such sequence. In symbols:

$$P(G, Ph) \approx \max_{GP \in S} \prod_{i=1}^{|GP|} P(GP_i | GP_0^{i-1}) \quad (1)$$

where *S* is the set of all complete graphoneme sequences *GP* such that the resulting string of graphemes is *G* and the resulting string of phonemes is *Ph*.

2.3. Unit Refinement

The initial unit selection algorithm results in graphonemes each consisting of a single phoneme. The graphonemes serve as the set of basic units for the ngram models used to estimate the joint likelihood of spelling-pronunciation pairs. However, these units may not be optimal for this task. In particular, they fail to exploit the predictable cooccurrence patterns of multiple phonemes generating a sequence of graphemes. We can remedy this shortcoming using algorithms developed for word phrase language models. We use the following iterative algorithm which is a slight simplification of that presented in [6]:

1. Sort the graphoneme pairs occurring in the corpus by bigram frequency.
2. Apply the joining operation to the *m* highest-ranking pairs in order.
3. Remove any joined units that fail a frequency criterion.

4. If no new units were added this iteration or NumUnits units have been added in total then halt; otherwise goto step 1.

Thus the first iteration will create graphonemes that consist of exactly two phonemes and later iterations can create graphonemes with longer phoneme sequences. Note that other criteria besides frequency have been used to select phrases in the literature with mixed results. This refinement in the set of graphonemes is followed by a step that removes graphonemes consisting of zero length grapheme sequences via concatenation using a one-step version of the above algorithm.

A Viterbi segmentation for the word "thoughtfulness" and its pronunciation in terms of these larger units is given in Figure 3.

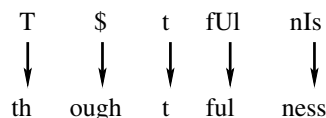


Figure 3. Segmentation of "thoughtfulness" and its pronunciation using refined graphoneme set.

3. Graphoneme Model Decoding

The estimated graphoneme model can be used to obtain an approximation to the joint probability of a spelling-pronunciation pair as given by equation (1). In order to estimate this probability efficiently we use a best-first multistack search algorithm similar to that described in [7]. The same basic algorithm is used whether we are searching for the pronunciation that maximizes the joint probability or whether there's a particular pronunciation of interest.

We maintain a separate fixed sized heap for hypotheses consisting of the most likely graphoneme sequences corresponding to grapheme strings of a given length. The heaps may be augmented with a beam threshold to further reduce computation. Initially, all heaps are empty except the zero-length grapheme string heap that has a single hypothesis consisting of "*<s>*". We then proceed iteratively. The most likely hypothesis is deleted from its heap and is extended by all appropriate graphonemes. Appropriate graphonemes are those that result in a grapheme string which is a prefix of the given spelling when appended to the current hypothesis. If we're interested in a particular pronunciation then extendable graphonemes must also result in a pronunciation prefix when appended. We then attempt to add these extended hypotheses to the heaps of the appropriate length and continue. If the current most likely hypothesis covers the entire word spelling, an extended hypothesis is created by adding "*</s>*". If this hypothesis already contains "*</s>*" then we have found the most likely graphoneme sequence corresponding to that word spelling (and pronunciation if provided).

4. Dictionary Verification

The dictionaries used in speech systems consist of many entries added manually, potentially by multiple experts. We would like to use the graphoneme to phoneme conversion model to flag suspicious dictionary entries. We can use the

graphoneme ngram model and the search algorithm described to efficiently estimate the likelihood of a given word spelling and its most likely pronunciation. We can also use these tools to estimate the likelihood of a given dictionary pronunciation for this word spelling. The ratio of these likelihoods is a measure of the relative likelihood of the dictionary pronunciation compared to the automatically generated one given the word spelling. This ratio is thus an appropriate measure to sort the dictionary entries by our confidence that they are errorful.

In order for meaningful likelihood ratios to be estimated, the words of interest must not have been included in the data used to train the graphoneme ngram model. We thus proceed in two steps. First the entire dictionary is segmented into graphoneme sequences as described in Sections 2.1 and 2.3. Then the dictionary is randomly partitioned into several parts so that a given word appears in one and only part. Then we go through each part, training the graphoneme ngram model on the remaining parts and applying it to the words in the heldout part, noting the likelihood ratios described.

5. Experiments

All experiments were conducted using the CELEX Lexical Database of English version 2.5 [8].

5.1. Pronunciation Guessing

In an effort to make comparisons with previously published results we proceeded as in [5]. All entries were lowercased and phrases and abbreviations were removed. There were the standard 26 grapheme symbols. The phoneme set consists of 53 symbols. The preprocessed database contains 66278 word forms. A random set of 40000 words was chosen for training and a disjoint random set of 15000 words was chosen for evaluation.¹

Given a reference and hypothesized pronunciation for a word, phoneme-level errors are assessed based on minimum edit distance. When multiple pronunciations for a word exist, a reference pronunciation resulting in the minimal number of errors is chosen. The phoneme error rate is then the number of phoneme-level errors divided by the number of phonemes in the reference pronunciations. The pronunciation error rate is simply the number of reference words for which the hypothesized pronunciation is not one of the reference pronunciations for that word, divided by the number of reference words.

In these experiments, we chose to vary only the number of multiphoneme graphonemes added in the unit refinement step and the order of the graphoneme ngram model estimated (previous experiments indicated that these were the most influential parameters). Note that the graphemeless graphoneme removal step is performed regardless of the number of multiphoneme graphonemes specified. In all cases, phoneme HMMs were limited to producing at most 5 graphemes each. In the unit refinement stage, 100 units were added per iteration with a minimum frequency of 25.

Performance results are summarized in Table 1. The general trend is the same for both phoneme and pronunciation error rate. For a given ngram order, the error rate drops as we

increase the number of multiphoneme units added until a minimum is reached and then the error rate begins to creep back upward. There is presumably a tradeoff between capturing correlations among multiple phoneme and grapheme sequences directly in single units and increasing the sparsity of the resulting ngram model. A similar trend has been observed with word phrase language models. The minimum phoneme error rate achieved is 3.68% (corresponding to a pronunciation error rate of 17.13%) with a 4-gram graphoneme model and no multiphoneme graphonemes added. Bisani and Ney [5] report a phoneme error rate of 4.02% on this evaluation set under identical training conditions. Thus our method slightly outperforms theirs and we may conclude that our algorithms are competitive.

NumUnits	NgramOrder	PhoneER(%)	PronER(%)
0	1	32.24	86.54
0	2	12.54	52.67
0	3	5.94	26.94
0	4	3.68	17.13
125	1	26.05	80.44
125	2	9.34	41.29
125	3	4.19	19.21
125	4	3.76	17.26
250	1	24.15	78.30
250	2	8.07	36.22
250	3	4.16	19.04
250	4	4.05	18.72
500	1	22.80	76.01
500	2	7.08	32.27
500	3	4.52	20.76
500	4	4.60	21.25

Table 1. Pronunciation guessing performance as a function of the number of multiphoneme units added and the order of the ngram model used.

5.2. Dictionary Verification

For this procedure, the preprocessed CELEX dictionary was split into 25 parts (i.e. 4% of the dictionary is heldout at a time). The phoneme HMMs were limited to producing at most 5 graphemes each. For the unit refinement stage, 100 units were added per iteration with a minimum frequency of 25 and 250 units were added in total. This resulted in an inventory of 1198 graphonemes. A trigram graphoneme model was used.

CELEX English 2.5	
PronER(%)	13.82
PhoneER(%)	2.86

Table 2. Pronunciation guessing performance accumulated over entire dictionary using cross-validation.

Pronunciation and phoneme error rates are reported on the entire dictionary, accumulating statistics for each of the 25

¹ The authors would like to thank Maximilian Bisani for providing the training and evaluation sets used in [5].

heldout parts. Table 2 summarizes the pronunciation guessing performance of the model on these datasets. Although the pronunciation error rate on the training dictionary is fairly high, figure 4 shows that most of the reference pronunciations are deemed relatively likely by the graphoneme model. Thus its application as a dictionary verification tool would seem promising.

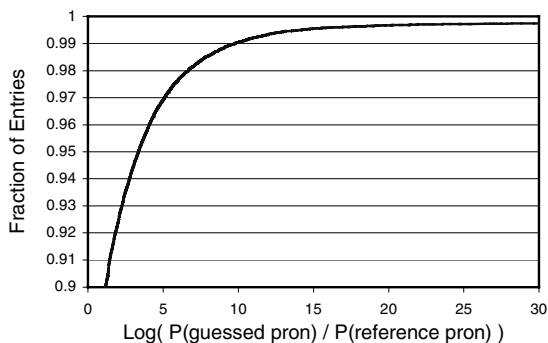


Figure 4. Distribution of conditional log odds of guessed and reference prons.

The 100 most suspicious dictionary entries as determined by the model were collected. Additionally a separate sample of 100 entries was randomly selected from the same dictionary. The two samples were combined and shuffled and presented to a human expert who was asked to note any discrepancies. 34 of the “suspicious” entries indeed required correction compared to only 1 for the random sample. Table 3 shows some spurious entries that were discovered in this manner along with the pronunciation suggested by our model.

Dictionary Spelling	Dictionary Pronunciation	Guessed Pronunciation
harrowing	h{r5rIN	h{r5IN
beseeching	bIsiJ	bIsiJIN
textured	tEksJ@R	tEksJ@d
sensitiveness	sEnslIv@tI	sEnS@tlvnIs
elusively	llusIv	llusIvI
capabilities	k1p@bII@tI	k1p@bII@tlz
effortlessness	Ef@tllsII	Ef@tllsnIs
workmanlike	w3m@nl2k	w3km@nl2k
comforter	kVf@t@R	kVmf@t@R
restaurant	rEst@r~N	rEstrQnt

Table 3. Errorful dictionary entries flagged by the dictionary verification procedure.

6. Conclusions and Future Work

We have presented a novel and competitive technique for data-driven language independent grapheme to phoneme conversion. The resulting framework assigns a likelihood to any word spelling-pronunciation pair allowing the validity of a dictionary entry to be assessed stochastically. This allows for

the automatic flagging of suspicious dictionary entries which can be presented to a human expert ranked by confidence.

We have augmented our internal dictionary-editing interface to automatically flag suspicious entries as they are entered using the described technique. Additionally, these models are used in our dictation software for assigning pronunciations to user added words.

In the future, we plan on considering techniques that attempt to select graphoneme units containing multiple phonemes and graphemes in a single step and compare them with our current technique on a common dataset.

7. References

- [1] Jelinek, F., *Statistical Methods for Speech Recognition*, The MIT Press, Cambridge, MA, 1997.
- [2] Luk, R.W.P. and Damper, R.I., “Stochastic Phonographic Transduction for English”, *Computer Speech and Language*, Vol. 10, pp. 133, 1996.
- [3] Bellegarda, J., “A Novel Approach To Unsupervised Grapheme-To-Phoneme Conversion”, *Proc. Pronunciation Modeling and Lexicon Adaptation for Spoken Language*, 2002.
- [4] Galescu, L. and Allen, J., “Bi-directional Conversion Between Graphemes and Phonemes Using a Joint N-gram Model”, *Proc. of the 4th ISCA Tutorial and Research Workshop on Speech Synthesis*, 2001.
- [5] Bisani, M. and Ney, H., “Investigation On Joint-Multigram Model For Grapheme-To-Phoneme Conversion”, *Proc. ICSLP*, pp.105, 2002.
- [6] Klakow, D., “Language-Model Optimization By Mapping Of Corpora”, *Proc. ICASSP*, pp.701, 1998.
- [7] Chelba, C. and Jelinek, F., “Recognition Performance of a Structured Language Model”, *Eurospeech*, 1999.
- [8] “CELEX lexical database”, <http://www.kun.ml/celex>.
- [9] Ney, H., Essen, U., and Kneser, R., “On Structuring probabilistic dependencies on stochastic language modeling”, *Computer Speech and Language*, vol. 8, no. 1, pp. 183, 1994.