REVISITING GRAPHEMES WITH INCREASING AMOUNTS OF DATA

Yun-Hsuan Sung[†*], *Thad Hughes*[*], *Françoise Beaufays*[*], *Brian Strope*[*]

[*] Google Inc., Mountain View CA
[†] Dept. of EE, Stanford University, Stanford CA

ABSTRACT

Letter units, or graphemes, have been reported in the literature as a surprisingly effective substitute to the more traditional phoneme units, at least in languages that enjoy a strong correspondence between pronunciation and orthography. For English however, where letter symbols have less acoustic consistency, previously reported results fell short of systems using highly-tuned pronunciation lexicons. Grapheme units simplify system design, but since graphemes map to a wider set of acoustic realizations than phonemes, we should expect grapheme-based acoustic models to require more training data to capture these variations.

In this paper, we compare the rate of improvement of grapheme and phoneme systems trained with datasets ranging from 450 to 1200 hours of speech. We consider various grapheme unit configurations, including using letter-specific, onset, and coda units. We show that the grapheme systems improve faster and, depending on the lexicon, reach or surpass the phoneme baselines with the largest training set.

Index Terms— Acoustic modeling, graphemes, directory assistance, speech recognition.

1. INTRODUCTION

Most large vocabulary speech recognition systems depend on three highly optimized models: a language model that estimates the probability of a sequence of words; a pronunciation model that describes how the words are divided into phoneme units; and an acoustic model that estimates the probability of observing a given acoustic feature vector in a given phonetic context.

While the language and acoustic models are typically trained with statistical training algorithms, the pronunciation models tend to be more ad hoc. Most commercial systems rely on a combination of a hand-made lexicon for common words and a pronunciation generation engine for words not listed in the lexicon. Often these pronunciations are later refined algorithmically based on acoustic data (e.g. [1]), or revised manually for increased accuracy.

While the language and acoustic models typically can grow and improve with more training data (e.g. more n-grams and longer spans for language models, more states and more Gaussians per state for acoustic models), the pronunciation models often don't scale well with increasing amounts of data.

This raises the question of whether it is desirable to keep a pronunciation model when large amounts of training data are available. In a sense, the lexicon provides a data-tying layer between the orthographic and acoustic representation of words, and as data increases, it is possible that this tying becomes unecessary and may even become a bottleneck.

One could easily build words out of letter-based units, or graphemes, instead of phoneme units, and transform the lexicon generation problem into a purely acoustic training problem. We may then expect common statistical approaches to lead to consistent improvements with increasing amounts of supervised and unsupervised data.

The idea of considering alternatives to phoneme units is not new. More than 20 years ago, Cravero et al. [2] proposed a unit set optimized for consistency and cardinality. Ten years ago, several research groups investigated syllable units, which have the promise of an improved mapping between spelling and acoustics [3, 4, 5, 6].

More recently and perhaps due to a growing interest for recognizing multiple languages, researchers confronted with the bewildering task of maintaining not one but several lexicons asked the inevitable question "what if we just used letter units instead?" Kanthak et al. [7] and Killer et al. [8] observed experimentally that for some languages, grapheme systems performed roughly as well as phoneme systems, but that for others, such as English, there was a high error-rate cost to moving to graphemes. This was attributed by the authors to the poor spelling to pronunciation correspondance of the English language, which is another way of observing that, in English, letter units lack acoustic consistency, and that consistency matters, much like Cravero et al. had suggested. But the experiments reported in these papers relied on training sets of roughly tens of hours of speech. If consistency matters, then the amount of data should matter too.

In this paper, we explore the scalability of grapheme systems, i.e. how quickly their performance improves with data, compared to phoneme systems. We base our experiments on data from GOOG-411 [9], an automated system that uses speech recognition and web search to help people call businesses. GOOG-411 is a good test bed for grapheme exper-

iments: business name recognition imposes interesting pronunciation and language modeling challenges, and a live commercial system provides complex acoustic variety.

2. PHONEME BASELINE SYSTEM

The speech recognition engine is a standard, large-vocabulary recognizer, with PLP features and LDA, GMM-based triphone HMMs with three states per triphone and 24 Gaussians per state, decision-tree state clustering, STC [10], and an FST-based search [11]. All acoustic models evaluated here are gender-independent, one-pass, and maximum-likelihood trained.

The lexicon used both for training and testing is a mix from various sources, with some manual tuning for entries that caused frequent recognition errors. A pronunciation engine trained from the lexicon using pronunciation by analogy (PbA) [12] is used as a backoff for words not in the lexicon. Some lexicon entries have multiple pronunciations, and PbA is configured to generate at most three pronunciations per word. The phone set consists of 43 Darpabet units. Sample lexicon entries are listed in Table 1.

word	pronunciation
apple	/ae/ /p/ /ax/ /l/
google	/g/ /uw/ /g/ /ax/ /l/
stanford	/s/ /t/ /ae/ /n/ /f/ /er/ /d/

Table 1. Lexicon entries in the baseline phoneme system.

3. GRAPHEME SYSTEMS

The grapheme systems described below are based on the same architecture as the baseline phoneme system, except that the unit set is different. The front-end, trainer, and decoder are unchanged. Context is still modeled by training tri-grapheme HMMs with 3 states per model. The decision-tree clustering algorithm uses a few broad "phonetic" classes adapted from true phonetic classes from the baseline system, e.g. *vowel: a e i o u*, *nasal: m n*, and the units themselves taken in isolation, e.g. *a: a*, *b: b*. No specific attempt was made at optimizing these classes; they are most similar to what Killer called "singletons" in [8].

3.1. 26-Letter Grapheme Systems

The first grapheme system we implemented uses the 26 letters of the English alphabet. Sample lexicon entries are listed in Table 2.

word	pronunciation
apple	/a/ /p/ /p/ /l/ /e/
google	/g/ /o/ /o/ /g/ /l/ /e/
stanford	/s/ /t/ /a/ /n/ /f/ /o/ /r/ /d/

Table 2. Lexicon entries in the 26-letter grapheme system.

3.2. Letter-Specific Units

To date, our training and recognition implementation does not support word-boundary context modeling, and isolated letters in acronyms are pronounced differently than within-word letters. Therefore, we included in the second grapheme systems a set of letter-specific units as shown in Table 3. These units are not as efficient as direct word-boundary modeling with decision trees, but at least preserve the context knowledge of the acronym during acoustic modeling. This brings the total number of units in this system to 52.

word	pronunciation
u	/_u_/
s	/_s_/
a	/_a_/
cat	/c/ /a/ /t/

Table 3. Lexicon entries in the grapheme system with letter-specific units.

3.3. Onset and Coda Units

Likewise, we added word-initial (onset) and word-final (coda) units in the third grapheme system, as shown in Table 4. Again, this makes relevant context information available for acoustic modeling. This grapheme system has 104 units.

word	pronunciation
apple	/_a/ /p/ /p/ /l/ /e_/
google	/_g/ /o/ /o/ /g/ /l/ /e_/
stanford	/_s/ /t/ /a/ /n/ /f/ /o/ /r/ /d_/

Table 4. Lexicon entries in the grapheme system with letter-specific and boundary units.

4. EXPERIMENTS

4.1. Data and Task

All experiments reported below were performed on GOOG-411 data. We defined four training sets of roughly 300K, 1M, 3M and 9M utterances (450, 1400, 4000, and 12000 hours) by picking random calls from our pool of manually transcribed data. These utterances contain city-state ("San Francisco California") and business queries ("Starbucks"), as well as commands ("go back", "start over"). The test set consists of 30K city-state and business utterances (no commands) taken from calls and calling periods not included in the training data.

The language model (LM) is a simple 100K phrase list that includes the test data transcriptions, and is placed in parallel with a 25K unigram containing all the words from the

phrase list. This is more manageable for rapid experimentation than the large production LM used for GOOG-411. By intentionally including the test data in the LM, we were able to approximate the error rate of the production system on this test set with a single small LM.

Performance is reported both in terms of word error rates and sentence semantic accuracy. In the latter, differences such as "kinko's" vs. "kinkos" or "italian restaurant" vs. "italian restaurants" are ignored in scoring.

4.2. Results

We first trained and evaluated a baseline phoneme system for each training set. The semantic-level sentence accuracy of these systems is reported in Fig. 1 (see the "Phoneme Baseline" curve). Accuracy increases by slightly over 1% absolute at each tripling of the training size, from 75.5% at 300K utterances to 78.3% at 9M.

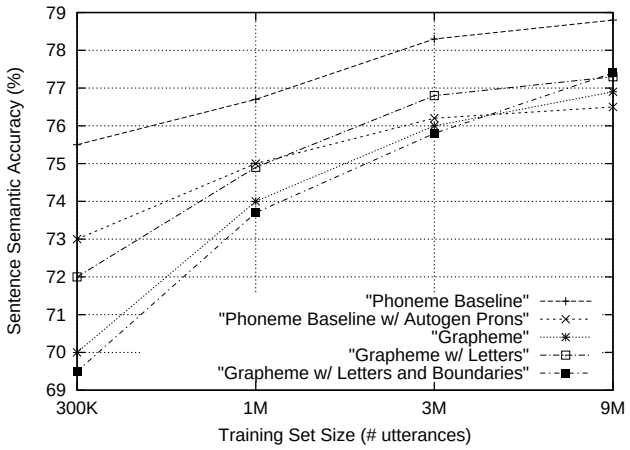

Fig. 1. Sentence semantic accuracy for the various systems.

Another phoneme baseline was then trained and evaluated by eliminating the pronunciation lexicon, thereby forcing all the pronunciations, in training and testing, to be autogenerated by the PbA pronunciation engine. This baseline is meant to give a sense of how much worse the phoneme system is when no (hand-tweaked) lexicon is available. Of course the PbA engine itself was trained from some lexicon, so this baseline does not totally eliminate the lexicon. The accuracy of this system, refered to as "Phoneme Baseline w/ Autogen Prons" in Fig. 1, is roughly 2% absolute worse than the "Phoneme Baseline" across the range of training set sizes, with 73% accuracy at 300K utterances to 76.5% at 9M.

We then trained and evaluated the grapheme systems. The first system, or "Grapheme" in the figure, with 26 letter units, starts 3% absolute lower than the "Phoneme Baseline w/ Autogen Prons" system for the smallest training set, but outperforms it as the amount of training data increases (76.9% vs 76.5% for the largest training set). This is consistent with Kanthak's and Killer's observations [7, 8] (Kanthak's English training set contained less than 100 hours of speech). It is also consistent with our intuition that training data can somewhat compensate for the acoustic diversity of English letters by implicitly modeling the various sounds corresponding to each letter symbol.

The second grapheme system, with letter-specific units, "Grapheme w/ Letters" in the figure, brings additional improvements over the simple grapheme models.

Finally, the full models with onset and coda units, "Grapheme w/ Letters and Boundaries" in the figure, show the most interesting behavior in terms of performance growth with data. This last system starts worst (69.5%) and ends best of the grapheme systems (77.4%): an 8% absolute gain as the data grows, compared to a 3.5% improvement for the baseline phoneme system. With 9M utterances, the largest training set we experimented with, the sentence semantic accuracy for the best grapheme system is within 1.4% of our baseline phoneme system. The last system doesn't work well with small amounts of training data because there aren't enough data to estimate parameters required by adding the extra units.

It should be noted that the systems compared here have roughly the same number of parameters: the grapheme system has more units (104 graphemes vs. 43 phonemes), but because the decision trees use the number of samples in a node as a split-stopping criterion, fewer tri-grapheme clusters are created on average per grapheme, resulting in roughly the same total number of states (18.1K states for the phoneme system, 18.3K for the grapheme system, with the 9M training set).

Fig. 2 shows the same analysis of the various systems, but this time considering word-error rates (WER). Using WER, the best grapheme system starts 5% absolute (19% relative) worse than the phoneme baseline with 300K training utterances, but with 9M utterances, the grapheme system is only 0.4% absolute (0.02% relative) worse than the phoneme system.

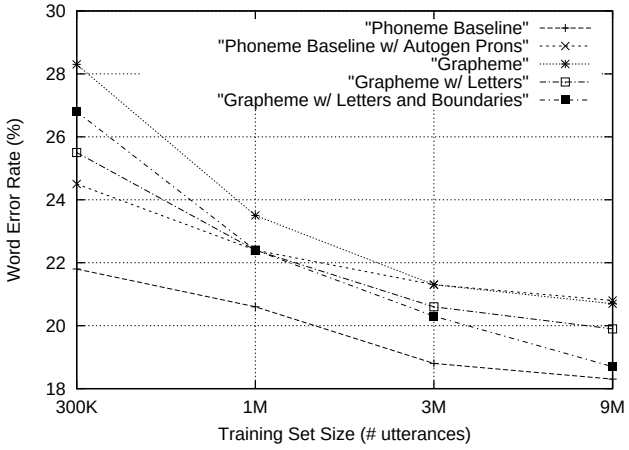

Fig. 2. Word error rate for the various systems.

While letter-units are a poor substitute to phoneme units for small systems, with increasing data and growing models, their performance improves faster.

4.3. Error Analysis

Table 5 compares some of the distributions of errors for the best grapheme system and the phoneme baseline. It shows different sub-sections of the test data and considers two signals: "OOL-utts" are utterances where the transcription includes at least one word that isn't in the lexicon so we used the PbA engine; and "LTR-utts" are utterances where the transcription includes at least one single-letter word (acronyms). $\mathcal{A}$ denotes the set of all utterances, $\mathcal{P}_c$, $\mathcal{P}_e$, $\mathcal{G}_c$, and $\mathcal{G}_e$ denote the sets of correct and error utterances for the phoneme and grapheme systems, respectively.

set	% utts	% OOL-utts	% LTR-utts
$\mathcal{A}$	100	3.7	5.6
$\mathcal{P}_e$	21.3	8.7	5.6
$\mathcal{G}_e$	22.7	4.1	7.4
$\mathcal{P}_e \cap \mathcal{G}_e$	18.3	8.1	5.7
$\mathcal{P}_e \cap \mathcal{G}_c$	3.0	12.1	4.5
$\mathcal{P}_c \cap \mathcal{G}_e$	4.5	4.3	18.3

Table 5. Percent sentence errors in various data subsets and systems (total = 30K sentences).

First, clearly most of the errors are common to both systems. While this limits system combination opportunities, it shows that with enough data and no lexicon, the grapheme system converges to mostly the same error distribution as the phoneme system.

Second, when the grapheme system corrected an error that the phoneme system made, the utterance is 3 times more likely than the average utterance to include a word that wasn't in the lexicon. This observation is consistent with the grapheme system being more accurate than the phoneme system with autogenerated pronunciations: graphemes are better than what are likely poor autogenerated pronunciations.

And third, when the grapheme system makes an error on an utterance that the phoneme system got right, the utterance is about 3 times more likely than the average utterance to include a single-letter word. While the letter-specific units provided improvements over the simple grapheme system, there is more to explore in terms of context, unit-selection, and data sharing.

5. CONCLUSION

We explored the feasability of replacing the phoneme units in a large-scale speech recognition system such as GOOG-411 with a set of letter-based units, thereby eliminating the need for a pronunciation lexicon and pronunciation engine, each of which imposes large off-line and run-time constaints on production systems.

We learned that with sufficient context modeling and enough training data, even with the orthographic-to-acoustic inconsistencies of English, graphemes may still be a suitable alternative to traditional phonemes. We saw comparable error rates with both systems, and graphemes seem to correct sentences with poor pronunciations. They seem to require proper modeling of word-boundary context, which we've only approximated through unit definition. Extending the unit set and context modeling may provide even faster improvements with increasing data.

6. ACKNOWLEDGEMENTS

This work was partially supported by the ONR (MURI award N000140510388).

7. REFERENCES

[1] F. Beaufays, A. Sankar, and M. Weintraub, "Learning linguistically valid pronunciations from acoustic data," in *Proc. Eurospeech*, 2003, pp. 2593–2596.

[2] M. Cravero, R. Pieraccini, and F. Raineri, "Definition and evaluation of phonetic units for speech recognition by hidden markov models," in *Proc. ICASSP*, 1986, pp. 42.3.1–42.3.4.

[3] R.J. Jones, S. Downey, and J.S. Mason, "Continuous speech recognition using syllables," in *Proc. Eurospeech*, 1997.

[4] S.-L. Wu, E.D. Kingsbury, N. Morgan, and S. Greenberg, "Incorporating information from syllable-length time scales into automatic speech recognition," in *Proc. ICASSP*, 1998, pp. 721–725.

[5] S. Greenberg, "Speaking in shorthand - a syllabic-centric perspective for understanding pronunciation variation," in *Proc. Esca Workshop MPV*, 1998, pp. 47–56.

[6] A. Ganapathiraju, J. Hamaker, J. Picone, M. Ordowski, and G. Doddington, "Syllable-based large vocabulary continuous speech recognition," in *IEEE Trans. on Speech and Audio Processing*, 2001, vol. 9.

[7] S. Kanthak and H. Ney, "Context-dependent acoustic modeling using graphemes for large vocabulary speech recognition," in *Proc. ICASSP*, 2002, pp. I.845–I.848.

[8] M. Killer, S. Stüker, and T. Schulz, "Grapheme based speech recognition," in *Proc Eurospeech*, 2003, pp. 4645–4648.

[9] M. Bacchiani, F. Beaufays, J. Schalkwyk, M. Schuster, and B. Strope, "Deploying goog-411: Early lessons in data, measurement and testing," in *Proc. ICASSP*, April 2008, pp. 5260–5263.

[10] M.J.F. Gales, "Semi-tied covariance matrices for hidden markov models," *Proc. IEEE Trans. SAP*, May 2000.

[11] "OpenFst Library," http://www.openfst.org.

[12] R.I. Damper and J.F.G. Eastmond, "Pronunciation by analogy: Impact of implementational choices on performance," in *Language and Speech*, 1997, vol. 40(1), pp. 1–23.