The AAAI-17 Workshop on
Crowdsourcing, Deep Learning, and Artificial Intelligence Agents
WS-17-07

Crowdsourcing the Pronunciation of Out-of-Vocabulary Words

**Sajad Shirali-Shahreza,‡ Pieter Luitjens, Natalie Morcos,
Wen Xiao, Zhenghong Qian, Gerald Penn***
Departments of Computer Science and ‡Mechanical and Industrial Engineering
University of Toronto

Abstract

We propose a method for crowdsourcing the pronunciation of out-of-vocabulary words, which in our experiments has generated a lexicon of competitive quality to the CMU pronunciation dictionary. In contrast to an earlier approach, we use crowdsource workers to generate new pronunciations, which are phonetically transcribed by an acoustic model, rather than merely to select among candidate alternatives from a letter-to-sound algorithm.

Introduction

Out-of-vocabulary (OOV) words still account for a significant number of the mistakes by both speech recognizers and text-to-speech synthesizers. These are not words that are merely very rare, but words that were unknown to the lexicon used by the automatic speech recognizer (ASR) or text-to-speech synthesizer (TTS). In the case of ASR, even if the pronunciation is accurately modelled, there can be a question as to how to spell it correctly. In the case of TTS systems, the pronunciation of the word may be unknown, as the component euphemistically known as "letter-to-sound" or "grapheme-to-phoneme" rules may in fact not be able to infer the pronunciation from the spelling of the word, particularly if its provenance is unknown, or the writing system is more logographically constructed.

What is less well appreciated is that, as OOV methods for ASR have matured into an area with highly specialized algorithms for OOV word prediction based on semantic vector representations (Horndasch et al. 2016), OOV part-of-speech tag prediction (Tafforeau et al. 2015), high-confidence (sub)phone-based OOV identification from speech (Karakos and Schwartz 2014; Lee, Tanaka, and Itoh 2016), recurrent OOV detection in speech corpora (Asami et al. 2016; Qin and Rudnicky 2013), topic-based OOV proper-name selection (Sheikh, Illina, and Fohr 2015), the area has moved very far away from OOV as it applies to TTS, where OOV detection and recurrence are a simple matter of string matching and meanings and POS tags, while not independent of pronunciation, fall well short of predicting it. The method we present here is a modern approach to OOV for the TTS domain.

A natural approach to guessing pronunciation without the acoustic analogue of the large amounts of fluent text that induce semantic representations in the ASR domain is to ask someone online how an unknown word should be pronounced. Crowdsourcing has in fact been explored as an option for this problem before by Rutherford et al. (2014), but their approach used crowdsourcing on a very small scale (10 repetitions from 10 crowdsource workers per keyphrase) to select among several hypotheses generated by a standard letter-to-sound algorithm using forced alignment. This is an extremely conservative use of crowdsourcing, particularly as their crowdsource workers really do speak the words in their experiments, rather than selecting the correct pronunciation in a multiple choice question format. Our approach uses nothing more than a larger number of speakers (101) and an acoustic model in order to find the pronunciation almost *ab nihilo*, by constructing phone lattices and submitting candidate pronunciation paths to a simple weighted voting algorithm that combines results across crowdsource workers. Our only assumption is that the basic phonetic inventory is known to the acoustic model (e.g., the pronunciation of *Rodriguez* selected using an English acoustic model would never trill the *r*'s). Furthermore, whereas Rutherford et al. (2014) experimented with popular proper names and neologisms sampled from entertainment-related queries to the Google Voice Search engine, we have experimented with a variety of words chosen according to a number of distributional properties. As a result, we are in a position to evaluate the use of our technique not only as a proxy for OOV entries augmenting an existing pronunciation dictionary, but also as a replacement for the entire dictionary — and with promising results. By using a larger number of crowdsource workers than the earlier attempt, and collecting votes among workers so that frequent, deprecated pronunciations can "gang up" to be selected, we can generate pronunciations that often differ from our reference, the CMU pronunciation dictionary, and yet are preferred by human judges on an almost equal basis.

An important altnernative is to search for pronunciations in written resources that are curated by the crowd such as *Wiktionary* (Schlippe, Ochs, and Schultz 2010).

The next section presents how the method works with

*This research was funded by an NSERC Strategic Project Grant.

some details on the acoustic model (which can be skipped, for the uninitiated). "Experiment" describes the data that were collected to test the method. "Results and Interpretation" then evaluates the method in four ways: an intrinsic string comparison with the CMU pronunciation dictionary, an examination of rank distribution in our n-best hypothesis lists, a preference test that uses human judges, and a comparison against a popular baseline.

Method

Contrary to Rutherford et al. (2014), our method does not require any other component of a speech recognizer than the acoustic model, but for convenience our acoustic model is built using the Kaldi toolkit (Povey et al. 2011) by treating each possible phone as a 1-phone word, with a trivial lexicon that maps every phone to itself, and a flat 0-gram language model.

We use Kaldi's Subspace Gaussian Mixture acoustic Model (SGMM), trained on 313 hours of Switchboard-I data (Godfrey and Holliman 1993), recorded in 8 kHz, 8-bit μ-law-encoded samples. The phone transition model was the default model for English that comes with Kaldi. This is an FST with an inventory of 43 phones that compactly represents 1509 biphones and 21 837 triphones. So there is in fact a bias towards English phonotactics in this model, in spite of the use of a uniform language model.

Digitization proceeds by computing cepstral-mean-and-variance normalised mel-frequency cepstral coefficients, split into 15 bins at 10ms intervals. We then apply linear discriminant analysis, followed by a maximum-likelihood linear transform (Gales 1998). For speaker adaptation, we use fMLLR (feature-scape Maximum Likelihood Linear Regression) (Gales 1999).

To decode candidate pronunciations, we generate phone lattices using the Exact Lattice method (Povey et al. 2012), then use Kaldi's n-best Viterbi decoder on the resulting phone lattices, with $n = 500$ and a search beam of 20%, the beam having been experimentally tuned on a development set with no overlap onto our test lexicon. Paths that collapse to the same phone representation are combined to boost the score of that representation.

Having obtained the 500 (or fewer) best pronunciations for each speaker for a given word, we rank them according to the score assigned by Kaldi, and calculate the score for each candidate pronunciation as:

$$s(j) = \sum_i (n - r_i^j + 1)$$

where $n = 500$ in our case, and r_i^j is the rank of the jth pronunciation in the n-best list of the ith speaker, or $n + 1$ if it is not in the top n. For example, the top-ranked pronunciation in the 500-best list for any speaker will contribute 500, while the 500^{th} best, when it exists, will contribute 1. We then select the pronunciation that has the highest total score $s(j)$ as the pronunciation of the unknown word.

Experiment

We evaluated our method by collecting an evaluation set of 100 words that are present in the CMU pronunciation dictionary, generating a pronunciation for them using the above method, and then comparing the generated pronunciation with those in the CMU dictionary.

Crowdsourced recordings

For each word, we sampled recordings from crowdsource workers on Amazon Mechanical Turk using an HTML 5 script that captured 2-channel, 16-bit linear-PCM audio samples at rates that varied between 44.1–48 kHz, depending on the browser. These were transcoded to 2-channel, 8-bit, 8 kHz μ-law samples, to match the training data of the acoustic model.

Unlike Rutherford et al. (2014), we sample one recording from about 100 speakers for each unknown word rather than 10 recordings from 10 speakers for each word. Rutherford et al. (2014) also trained on 7 of the 10 utterances, using the other 3 for evaluation, whereas our recordings were only used for testing. We also made no attempt to vett our workers for voice quality, speaking rate or other factors pertaining to speakers, apart from having them self-identify as native speakers of North American English, a point that Rutherford et al. (2014) is underspecific on.

We did, on the other hand, filter our sample of 101 crowdsource workers according to the acoustic quality of their recordings, by randomly selecting two words from the evaluation lexicon and subjecting their recordings by every speaker to a variety of acoustic tests, plus manual confirmation of foreign-speech accentuation. For simplicity, we interpreted negative results of these tests not as evidence of bad recordings, but as evidence of bad crowdsource workers, and thus discarded all of their recordings. This interpretation is based upon the premise that a worker that cannot record two words well is probably incapable of recording any words well.

The acoustic tests, conducted by hand by a single engineer, looked for the presence of:

1. echo or reverb (31)
2. actuator clicks (23)
3. non-linear distortion (related to the microphone or codec) (23)
4. background noise or microphone disturbance (10)
5. foreign-accented speech (9)
6. silence (no speech recorded) (5)
7. cutoff (2)
8. clipping (1)
9. stammering/repetition (1)
10. other recording errors (1)

The number of crowdsource workers who failed each test is given in parentheses. Every worker was classified into one of three groups: good (17) if both samples failed no tests, borderline (26) if one or both samples failed one test, and bad (58) if one or both samples failed two or more tests. Note the very high number of bad workers; a reasonably clean recording of two isolated words of speech was beyond the technical competence of over half of our sample, and the use of their

recordings in our method caused a significant degradation in performance. We experimented with using the recordings of the good workers, as well as with using those of both the good and borderline workers, on the other hand, and found that the results were nearly identical. What we report here are the results of using both good and borderline workers, of whom there were 43.

Experimental OOV Lexicon

To greatly simplify the evaluation, words with known pronunciations were selected as putative OOV words for the purposes of this experiment. These words were sampled from the CMU pronunciation dictionary according to several criteria that were formulated: (1) to create a balance among the lexical items so that they could together be construed as representative of the English lexicon as a whole, and (2) to characterize the properties of OOV words as a whole. By creating distributions along these criteria, we may then explore them for potential weaknesses in the method. One of our criteria is token frequency, for example, but exploring the rare side of this distribution is merely one of the approximations that we have at our disposal for investigating the properties of OOV words.

Our criteria are:

1. Word frequency: Words were selected across 50 evenly spaced frequency bands, divided geometrically between 0 and 12.811974785439626 in the natural log domain. Two words were sampled from each frequency band.
2. Word length: Words were sampled so as to adhere to a power-law distribution in their length, with a log-domain slope of -0.59666587 and a log-domain y-intercept of 15.96348835. These parameters were determined from a log-domain least-squares fit to English with an R-value of 0.92323035. The R-value of our obtained sample to this curve is 0.7852613399.
3. Monophone entropy: Words were sampled so as to resemble the overall discrete entropy of the distribution of the 43 phones of Kaldi's phone transition model. In English, we estimated this at 3.33231165807. We obtained a discrete entropy of 3.34432636829 for our sample.
4. Number of ambiguously pronounced words: 11.84% of the CMU pronunciation dictionary has more than one pronunciation. 12% of our sample does.

All benchmarks for English were derived from the American National Corpus. The resulting evaluation lexicon is shown in Figure 1.

Results and Interpretation

There are a number of ways to evaluate the proposed method.

1-Best distance

The first is simply to accept the CMU pronunciation dictionary as the gold standard, and measure our choices' deviation from that. The macro-averaged Levenshtein distance between the pronunciation selected by the above method and those of the CMU dictionary is $0.269078(\sigma = 0.229368)$. Macro-averaging is necessary because there are multiple pronunciations for some words in the CMU pronunciation dictionary; the micro-averaging results are very similar, however.

Likelihood

The second way is again to accept the CMU pronunciation dictionary as a gold standard, but to examine the rank of CMU pronunciations within the 500-best lists of our crowdsource workers. This is a non-parametric cousin of a data likelihood computation in a probabilistic model. In the case of 20 of the 100 words, our method selected a pronunciation found for that word in the CMU dictionary. A median of 4 (9.3023%, $\sigma = 9.562426$) of the 43 crowdsource workers generated the CMU pronunciation somewhere in their 500-best list, with each worker generating the CMU pronunciation a median of 20 times among the 100 words ($\sigma = 7.74968295$). In the case of 30 words, not even one crowdsource worker's n-best list generated a CMU pronunciation. Among the other 70, the rank of the CMU pronunciation was a macro-average of 38.4698 (min=1, max=210) across all words, out of a possible range of $1 - -500$. Note that not all lattices generated 500 paths, however; the median number of paths was 280 ($\sigma = 203.387$). Figure 2 shows the distribution of ranks and their medians for each word in the lexicon.

The winning pronunciation for 80 of the 100 words was one not found in the CMU dictionary. An average of 15 (34.8837%, $\sigma = 7.798446$) of the 43 crowdsource workers generated the winning pronunciation somewhere in their 500-best list, with each worker generating the winning pronunciation an average of 37 times among the 100 words ($\sigma = 14.40988$). In the case of only one word (CATAPULTING) did no speaker generate it in their 500 best.

From these results, we can conclude that a setting of n that is substantially lower than 200 would not fare well without further improvement on either our selection method or the quality of crowdsource work obtained. The average rank of 38 could have been far worse, but CMU pronunciations are too often entirely absent (30%), and too many workers (about 90%) are not generating them. Reducing n below the rank at which the "correct" CMU answer occurs would increase both of these percentages.

Human-subject evaluation

The third way is to consider all pronunciations as candidates, regardless of source, and ask crowdsource workers to evaluate them. We evaluated our selected pronunciations and the CMU dictionary's pronunciations in two ways: by asking crowdsource workers to transcribe synthesized pronunciations, and by asking crowdsource workers to choose which of two alternative pronunciations (ours and one of the CMU pronunciations) is better.

185 transcribers/voters were recruited, again on Amazon Mechanical Turk. Crowdsource workers who had participated in the collection of acoustic data were ineligible. Of these, 145 completed both the transcription and selection tasks, 39 resigned during transcription, and 1 resigned during selection. The prompts presented to the work-

RATION	SWAYZE	LUCIANO	SCULPTURES
RIVALS	HAYLEY	YASSIN	LAPSES
GASPING	RADIOED	HUTCHINS	FLOPPY
RENAUD	CAUTIONARY	MURATA	ILIESCU
ASTUTE	FLOYD	LOOP	PERMEATING
RILE	INSINUATE	UNHEEDED	HACK
BUNDLED	WINGERS	POTVIN	REROUTING
ANNUALIZED	RECESSIONARY	GABBY	SHOWCASING
GETS	BOERNER	EDU	BECKMAN
INHOSPITABLE	WATCHDOG	PENTAGON	JULIET
TURQUOISE	CONVICTION	DIONYSIUS	BARRAGE
NOV	APPRAISE	CREDIBLE	KNAUS
MIDDLEMAN	NEWTS	ENDOMETRIOSIS	LIBERALISM
FREAK	INVINCIBLE	ROMER	JOSTLE
SOHO	CATAPULTING	UNDRESSED	VAUGHAN
CROCKETT	INTERVENTIONIST	INVADES	SASSOU
NONGOVERNMENTAL	DOCUMENTARIES	BREMER	CARJACKINGS
SHARPE	WASP	UNHAPPY	SMOOTHING
BAFFLE	MATTHEW	CHANDELIERS	DOUSE
OUTCASTS	HUGE	NORTHERNERS	SHENG
CLAUDE	BRAVO	ENVIRONMENTALISTS	SHORTFALLS
RACY	HOURS	GIZMOS	CLASP
GOOF	RESPECTING	UNSEATING	HENDRY
SUPERVISION	CONGRESSES	RESENTED	FATAH
ANGEL	WITHHOLDS	INTERROGATORS	INCARNATIONS

Figure 1: The experimental OOV lexicon.

ers were synthesized in Festival using the `us1_mbrola` voice. Workers could only listen to each prompt once, and only in the order presented.

The transcriptions of CMU pronunciations had an average Levenshtein distance of $0.28944 (\sigma = 0.35723)$ whereas those of pronunciations generated by the experimental method has an average distance of $0.39175 (\sigma = 0.41806)$, suggesting that the CMU pronunciations were only modestly more comprehensible because of the high standard deviations.

As for the selection task, workers were asked to choose whether: (1) the first of two prompts was better, (2) the second was better, or (3) the two were about the same. CMU and experimental pronunciations were evenly permuted between presentations as the first or second choice to control for presentation bias.[1]

In addition to reporting who got more votes, we can also calculate a *differential percentage*, which measures the margin by which the vote carried as a percentage of the total number of votes cast. As there were three outcomes available to voters, this is calculated as:

$$D = \frac{E - C}{E + C + I}$$

where E is the number of votes for the experimental pronunciation, C is the number of votes for a CMU pronunciation, and I is the number of votes that expressed no preference. Thus D is negative when a CMU pronunciation wins, and positive when an experimental pronunciation wins.

The CMU pronunciation received more votes on 42 word comparisons (average $D = -45.6463\%$, $\sigma = 23.18238$), the experimental pronunciation received more votes on 38 word comparisons (average $D = 29.05185\%$, $\sigma = 22.17113$), and there were no ties[2] The other 20 words are those for which the two pronunciations were segmentally identical. It is important to note, however, that the CMU pronunciation dictionary also contains stress accent annotations, which our method does not generate. We synthesized CMU pronunciations with the stress indicated, and so there is a bias towards CMU pronunciations because of their presumably more natural stress accentuation than whatever Festival guesses by default.

Over all 80 words with non-identical pronunciations, the CMU pronunciations received 2693 votes (46.26%), our experimental pronunciations received 2113 votes (36.3%), and 1015 votes (17.44%) were cast for pronunciation pairs being indistinguishable. The average D over all words was -10.1647%, with $\sigma = 43.79747$.

What we can infer from these results is that, while the CMU pronunciations do not win significantly more often, or by a significant margin on average ($D = -10.1647\%$), they do tend to win by larger margins ($D = -45.6463\%$) when they win at all.

[1] We did confirm the presence of presentation bias: voters were 1.43 times more likely to choose the second choice than the first, and did so for 80% of the words. Even among cases where the winning method (CMU) was the first choice, the second choice was still 1.15 times more likely to be chosen.

[2] That is, no word received as many preferential votes for one pronunciation as for the other; there were 1015 votes of no preference.

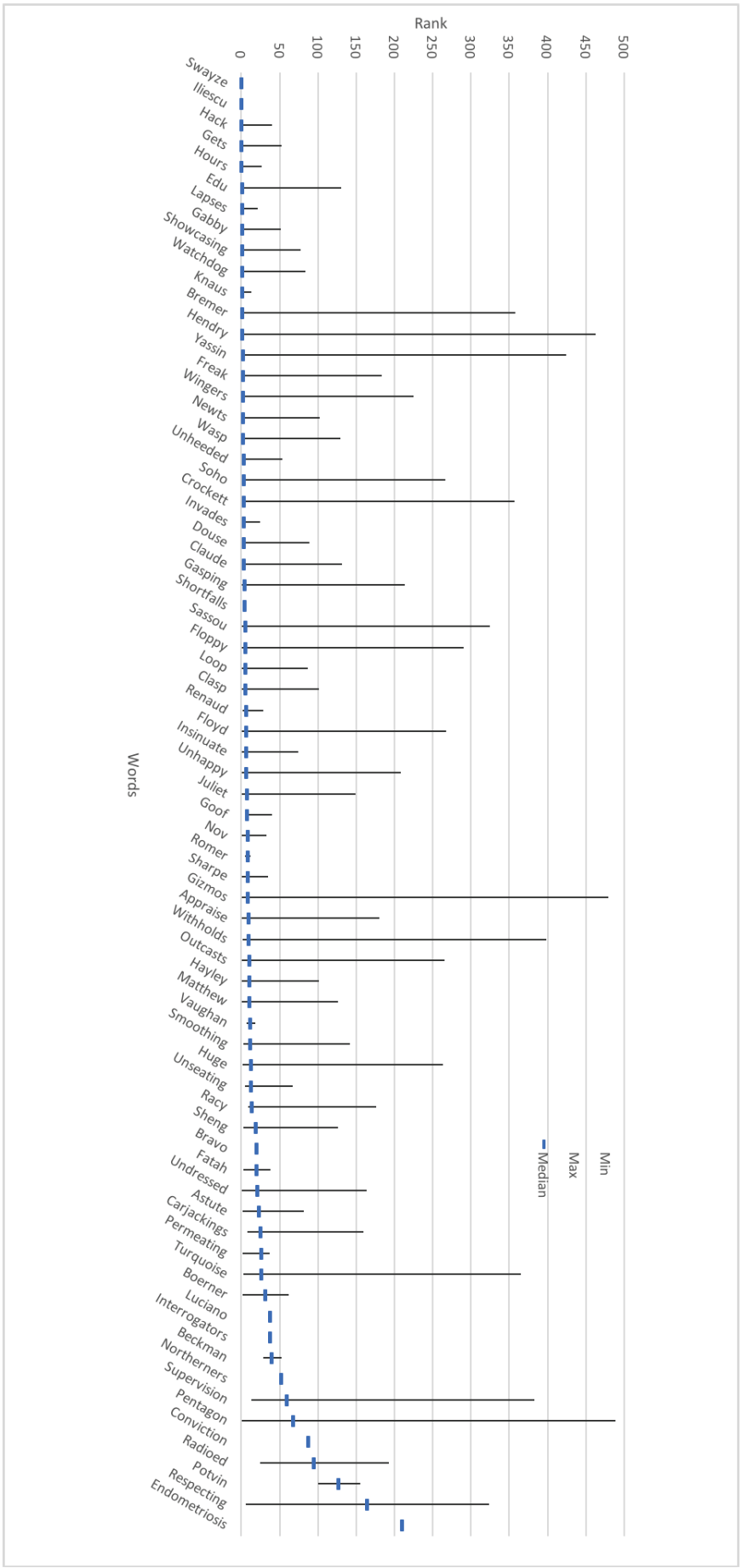

Figure 2: Median, minimum and maximum ranks of the CMU pronunciations, ordered by median rank. A lower rank number is better.

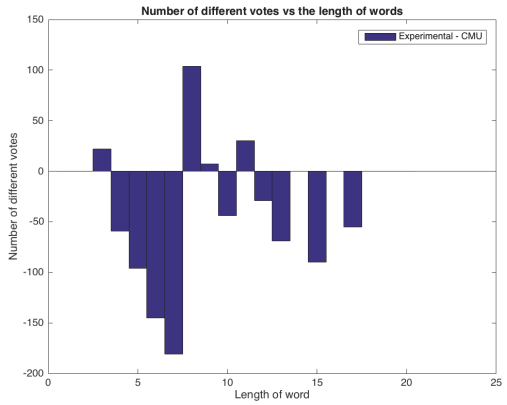

Figure 3: The difference in votes as a function of word length.

On the whole, this is still a remarkably positive showing for a method that uses no letter-to-sound productions, and in which roughly two thirds of the crowdsource workers generally do not even generate the pronunciation selected. The reason is our selection procedure, which cumulatively *adds* subtracted ranks within 500-best candidate lists, and our setting of $n = 500$. The crowd does very well for itself in aggregate, even though no individual worker does. While our pronunciations are often different, they are considered very good by human judges — better than CMU's almost half the time.

There is, on the other hand, no apparent correlation between Levenshtein distance in the transcription task and user preference in the selection task, nor between performance in any of these three evaluations and the distributional criteria that we had identified for sampling the lexicon. Figure 3, for example, shows the distribution in the difference of votes $(E - C)$ as a function of length. This has a Pearson sample correlation coefficient of 0.0198, with $p = 0.4812$ using a permutation significance test over 10^6 sampled reorderings. The other factors are similarly uncorrelated.

Baseline

Finally, we compare our approach to what still remains the most common baseline for OOV pronunciations in the TTS community, the decision-tree-based approach implemented in the Festival TTS system (Black, Lenzo, and Pagel 1998), which learns rules for assigning an allowable phone to each letter in the orthographic input. Out of the box, this approach fares very well on our 100-word sample, because it was trained on the CMU pronunciation dictionary, which contains our sample. We retrained this system on a subset of the CMU dictionary, excluding our sample words and all words containing any of our sample words. The retrained Festival LTS system performs at 90.4184% accuracy, which is far superior to one minus our macro-averaged Levenshtein distance of $0.269078 = 0.730922$. But of the 51 phone instances wrongly predicted by the LTS component,[3] our experimental method correctly guessed 32 of them (62.7451%), guessing the same (and incorrectly) as LTS in only 6 cases (11.7647%). One of the present authors furthermore evaluated the remaining 13 cases and determined that the incorrect pronunciation assigned by our experimental method was clearly better than the incorrect pronunciation assigned by the Festival LTS component in 9 of them (17.6471% of all 51 cases).

What is more interesting is that 22 (68.75%) of the 32 cases in which the experimental method corrects Festival's LTS component are vowels, and all but one (in which Festival wrongly predicted a continuant) are continuants. This is perhaps unsurprising, as continuants have more energy and so can generally be predicted from speech more easily, but this complementarity will clearly be very useful for designing a hybrid of these two approaches, a task that we have yet to undertake.

Conclusion

The proposed method for selecting pronunciations of OOV words has proved to be competitive with the phone sequences found in the CMU pronunciation dictionary, across several different distributional criteria, even though there is not a great deal of overlap between the two sets of pronunciations.

In addition to the hybrid-LTS method mentioned above, possible extensions of this method for future research are:

- the use of alternative vote combination functions than adding subtracted ranks,
- the addition of stress accent prediction, which at present hampers our ability to fairly evaluate against the CMU dictionary,
- the automation of most or all of the acoustic tests to render our method completely automatic,
- the use of higher sampling rates and sizes than the Switchboard-I acoustic model that was used in this experiment,
- better feedback and priming of crowdsource workers to lower the substantial attrition rate that we experienced when evaluating the quality of the recordings they submitted, and
- integration with recent work on pronunciation verification and quality classifiers (Rao, Peng, and Beaufays 2015) to improve performance.

References

Asami, T.; Masumura, R.; Aono, Y.; and Shinoda, K. 2016. Recurrent out-of-vocabulary word detection using distribution of features. In *Proc. INTERSPEECH*, 1320–1324.

Black, A.; Lenzo, K.; and Pagel, V. 1998. Issues in building general letter to sound rules. In *Proc. 3rd ESCA Workshop on Speech Synthesis*, 77–80.

[3]There were 70 instances in total, but 19 involved epsilon predictions that are harder to ascribe to actual pronunciation differences versus alignment parallax.

Gales, M. 1998. Maximum likelihood linear transformations for HMM-based speech recognition. *Computer Speech & Language* 12(2):75–98.

Gales, M. 1999. Semi-tied covariance matrices for hidden Markov models. *IEEE Transactions on Speech and Audio Processing* 7(3):272–281.

Godfrey, J., and Holliman, E. 1993. Switchboard-1 release 2. Philadelphia: Linguistic Data Consortium. LDC97S62.

Horndasch, A.; Batliner, A.; Kaufhold, C.; and Nth, E. 2016. Combining semantic word classes and sub-word unit speech recognition for robust oov detection. In *Proc. INTERSPEECH*, 1335–1339.

Karakos, D., and Schwartz, R. 2014. Subword and phonetic search for detecting out-of-vocabulary keywords. In *Proc. INTERSPEECH*, 2469–2473.

Lee, S.-W.; Tanaka, K.; and Itoh, Y. 2016. Generating complementary acoustic model spaces in dnn-based sequence-to-frame dtw scheme for out-of-vocabulary spoken term detection. In *Proc. INTERSPEECH*, 755–759.

Povey, D.; Ghoshal, A.; Boulianne, G.; Burget, L.; Glembek, O.; Goel, N.; Hannemann, M.; Motlíček, P.; Qian, Y.; Schwarz, P.; Silovský, J.; Stemmer, G.; and Veselý, K. 2011. The Kaldi speech recognition toolkit. http://www.danielpovey.com/files/2011_asru_kaldi.pdf.

Povey, D.; Hannemann, M.; Boulianne, G.; Burget, L.; Ghoshal, A.; Janda, M.; Karafiát, M.; Kombrink, S.; P. Motlíček, P.; Qian, Y.; Riedhammer, K.; Veselý, K.; and Vu, N. 2012. Generating exact lattices in the WFST framework. In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 4213–4216.

Qin, L., and Rudnicky, A. 2013. Finding recurrent out-of-vocabulary words. In *Proc. INTERSPEECH*, 2242–2246.

Rao, K.; Peng, F.; and Beaufays, F. 2015. Automatic pronunciation verification for speech recognition. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 5162–5166.

Rutherford, A. T.; Peng, F.; and Beaufays, F. 2014. Pronunciation learning for named-entities through crowd-sourcing. In *Proceedings of Interspeech 2014*, 1448–1452.

Schlippe, T.; Ochs, S.; and Schultz, T. 2010. Wiktionary as a source for automatic pronunciation extraction. In *Proc. INTERSPEECH*, 2290–2293.

Sheikh, I.; Illina, I.; and Fohr, D. 2015. Study of entity-topic models for oov proper name retrieval. In *Proc. INTERSPEECH*, 1344–1348.

Tafforeau, J.; Artieres, T.; Favre, B.; and Bechet, F. 2015. Adapting lexical representation and oov handling from written to spoken language with word embedding. In *Proc. INTERSPEECH*, 1408–1412.