UTTERANCE CLASSIFICATION IN SPEECH-TO-SPEECH TRANSLATION FOR ZERO-RESOURCE LANGUAGES IN THE HOSPITAL ADMINISTRATION DOMAIN

Lara J. Martin, Andrew Wilkinson, Sai Sumanth Miryala, Vivian Robison, and Alan W Black

Carnegie Mellon University
Language Technologies Institute
5000 Forbes Avenue, Pittsburgh, PA

ABSTRACT

Although substantial progress has been achieved in speech-to-speech translation systems over the last few years, such systems still require that the speech be written in some appropriate orthography. As speech may differ greatly from the standardized written form of a language, it can be non-trivial to collect written data when there is no standard way for it to be represented. This project addresses the problem from the other end and expects that speech alone is available in the target language, and that no (standard or non-standard) orthography exists. It, therefore, treats the acoustic representation of the language as primary and uses language-independent methods to produce a phonetically-related symbolic representation that is then used in the translation system. Thus, the speech translation system is created for the target language as defined by the recording of that language rather than some body of orthographic transcripts.

In this work, we are creating an application called APT (Acoustic Patient Translator), which uses a novel scheme of speech recognition and translation within a targeted domain. By working with a set of predefined sentences appropriately chosen to fit a scenario, we use utterance classification as a speech recognition algorithm. The utterance classification is achieved using cross-lingual, language-independent phonetic labeling. Since we are working with a set of select phrases, the translation part is trivial. We are concentrating on communication with hospital staff, such as scheduling a doctor's appointment, as our domain. In addition to English, we also run experiments on Tamil.

***Index Terms*—** zero-resource languages, speech-to-speech translation, utterance classification, speech recognition

1. INTRODUCTION

There are many languages in the world which are severely lacking in language technologies. Although a lot of work is being done for languages such as English and Spanish, very little is done towards speech technology for languages with fewer resources. This may be because of the limited data available or a relatively much smaller number of speakers. This work concerns building speech technology and translation systems for languages where only limited audio is available and there is no readily-available or well-defined writing system. Since there is no text data available, we cannot easily train standard language models, or convert speech to text, thus making the task of translation quite different.

Some earlier speech-to-speech translation systems have addressed the issue of there being no recognized written script by defining one. Kathol *et al.* [1] developed a Romanized writing system for Pashto, but admitted that training people to use it consistently was a limitation when collecting later data. Within the DARPA TransTac program, inconsistencies were noted in word representation choices in the defined orthography of the Iraqi Arabic dialect [2]. We take a different route, by discovering a data-driven, phonetically-directed symbolic representation of the acoustics. We then use a classification technique to distinguish between each utterance type. At some level this is similar to Voxtech's Phraselator [3], but we have a two-way system and do not require a language-specific speech recognizer.

Stüker and Waibel [4] present experiments toward extracting wordlike sequences and bilingual alignments from phonetic representations of utterances that could be generated using bilingual speakers. Their system relies on ideal transcriptions, and is based on a preexisting corpus.

We are developing a way to make a language-independent dialogue system which can easily support new languages without any linguistic knowledge or expertise of the user. All that is needed is a set of recordings dynamically generated from a few speakers.

We intend to simplify this problem by only selecting a small set of phrases pertinent to a topic of conversation. In this project, we picked the topic of administrative processes surrounding a doctor's appointment. The conversation is between a non–English-speaking immigrant and an administrative staff member at an American hospital. For initial work, we further simplify the problem by using data from a select demographic. We utilize cross-lingual phone-recognition technologies to find phonetically inspired symbolic representations of speech. This work is in collaboration with University of Pittsburgh Medical Center (UPMC) Montefiore.

2. THE PHONE APPLICATION

We are developing a smartphone application with three main functions. First, the app allows a bilingual speaker of both a zero-resource language—the source language—and English to read the English phrases and record their spoken translations. Second, it allows other source language speakers to listen to each recorded phrase and to record themselves repeating each phrase. These two functions together will enable users to easily supply additional languages. Third, by applying our research methodology to the collected data, it will allow an English-speaking staff member and a non–English-speaking patient to hold a turn-based conversation by recognizing the phrase one user speaks and playing the corresponding translated phrase in the other user's language. The app is being developed for Android, and we intend to make it available for download for a variety of smartphones and tablets. This way, the hospital staff will not need any specialized equipment to use the interface.

3. THE CORPUS

In establishing a corpus of phrases of sufficient size and breadth to enable conversation in the domain, several cultural factors must be considered. The phrases, though we devise them in English, should translate well into other languages and be relevant to other cultures. However, phrases like "Is she a minor?" and "Are you her legal guardian?" do not translate well to cultures in which the concepts of "minor" or "legal guardian" do not exist or are defined differently from the American legal system. Regardless, these are questions that the hospital staff are expected to ask, and therefore they should be accounted for.

With this in mind, we began with an initial set of 102 phrases intuitively relevant to the domain, standing in for a more expanded and fine-tuned set to be constructed later. Examples include: "I'd like to reschedule my appointment," "Do you have insurance?" and "What is the patient's date of birth?".

We interviewed two staff members at UPMC Montefiore who regularly engage with non–English-speaking patients. We based model conversations on our initial phrases, constructing interactive scenarios in which our respondents provided examples of their typical dialogue during such interactions. Similar methods with standardized scenarios have been used to create a corpus of patient-doctor interactions [5]. We identified the most common and useful statements resulting from the interviews. Using these as a guide, we wrote somewhat more general phrases that captured the meaning of the originals. Each original statement maps to one or more general phrase semantically. There are currently around 750 phrases in the list. For many of the statements, we wrote one or more paraphrases—other ways that the same semantic meaning could plausibly be conveyed—that our system can recognize as alternatives of the same basic statement. We have not yet collected audio data for all the phrases; experiments are conducted using the original 102-phrase list.

The subjects we used for data gathering are played a series of audio prompts and asked to repeat them. This is necessary for the "zero-resource" scenario in which no standard written form exists for the source language, and also for collecting spoken data from illiterate people who would not be able to read a textual prompt even if one was available. Therefore, our mobile app makes the recording process easy for the user.

3.1. Languages

All recorded speakers are young adults (20-35 years old), and are native speakers of Tamil or American English. There are 12 American native English speakers (5 male and 7 female) and 5 male native Tamil speakers. We also perform experiments with 3 of the American English speakers (2 male, 1 female). We chose the non-English language owing to ease of access to speakers, as well as for being distinctly different from English. All of the speakers of Tamil are also fluent in English, but we acknowledge and expect that end users of the app may not know any English.

3.2. Collection

For each language, we designated one speaker to provide the recordings that the later speakers would hear as prompts. For Tamil, these initial speakers first translated the English phrases into their language before recording themselves speaking.

The audio was recorded in 16-kHz mono using our app interface running on various phones and tablets. Silences at the beginnings and ends of sound files were trimmed using Audacity's "Truncate Silence" tool. Participants were informed of the purpose and application of the recordings.

4. METHODS

After concluding the recording step, the datasets on which we experiment consist of, for each language, recordings made by each speaker of all 102 phrases in that language. (For clarity, we refer to a "phrase" as one of the 102 sentences, and an "utterance" as a recorded instance of the phrase said by one of the speakers.) The overall goal of our experiments is to be able to consistently and accurately match a given utterance from a held-out speaker to the same utterance as spoken by the non–held-out speakers. With one speaker at a time held out, we train on the rest of the speakers and use the held-out speaker for testing (leave-one-speaker-out cross-validation). The accuracies that we report are from averaging over the values produced from each held-out speaker.

By optimizing performance of the classification through training on our datasets, we aim to be able to perform accurate classification of utterances spoken by end users during a real-time dialogue. In pursuit of optimizing classification, we experimented with a variety of representations of the utterances and types of metrics for computing string edit distance (SED) between them.

4.1. MFCC Dynamic Time Warping

We began by transcribing all utterances as vectors of mel-frequency cepstral coefficients (MFCCs). To compute similarity between utterances in this representation, we used dynamic time warping (DTW). For each held-out speaker, for each utterance, we compute the mean and standard deviation (SD) of the pairwise comparisons between non–held-out speakers; then for each utterance we compute the mean of the pairwise comparisons between the held-out speaker and each non–held-out speaker, compute the z-score for that utterance, and return a tuple of the utterance number and the z-score. We sort the tuples by increasing z-score (lower is better), and record whether the original (correct) utterance number is in the top 1, top 5, or top 10 results and record the mean rank of the correct utterance.

This process has the benefit of being language-independent, since the method of representing the data as MFCCs is directly acoustically derived and takes no language-specific information into account. The drawback is that it is very slow and computationally expensive; also, the accuracy in our experiments was low. Therefore, we investigated more high-level approaches that use symbolic representations. The results for this baseline are shown in Table 1.

Experiment	% Top 1	% Top 5	% Top 10	Avg. Rank
DTW-Pruned	7.108	16.585	21.814	57.453

Table 1. Results for English Experiments with MFCCs

4.2. Features

Due to the expense and limited success of the MFCC approach, we investigated using faster, symbolic representations. We used two types of automatically-extracted symbolic features: English phonemes and language-independent inferred phones (IPs).

4.2.1. English Phonemes

We began by creating a phonemic language model of English. This model is not trained on English words, but rather on the phonemic sequences that compose words. We trained the model on a corpus of transcribed TED talks that we converted into Arpabet notation using the Carnegie Mellon University Pronouncing Dictionary. Then, using the work of Sitaram *et al.* [6], we predicted English phonemes from our audio.

An example from our English data:
Original Phrase: "What brings you here today?"
Eng. Phones: "SIL W AH T P R IH NG Z IY HH IH R D IH D EY"

For our initial work, we use only one phone recognizer, trained on English, since we will be translating to and from English. However, we intend to use multiple phone recognizers trained on different languages to make the phone recognition step more robust.

4.2.2. Inferred Phones

After predicting and outputting phonemic transcriptions of the utterances, we extracted "Inferred Phones," or IPs. The IPs are produced by creating a tree of clustered groups of sounds which share similar articulatory features [7]. We created several trees for each language that varied in the number of IPs. Due to the nature of the tree creation, we cannot specify exact numbers of IPs; but we can produce configurations in a "ballpark" range. These configurations are:

English: 15, 17, 30, 47, 84, 92, 101, 123, 171 IPs (12 speakers).
Tamil: 14, 17, 24, 51, 82, 93, 103, 118, 165 IPs (5 speakers).

An example from our English data:
Original Sentence: "What brings you here today?"
47 IPs: "ip25 ip26 ip26 ip26 ip25 ip4 ip13 ip13 ip13 ip13 ip14 ip14 ip14 ip14 ip24 ip25 ip24 ip28 ip21 ip17 ip15 ip15 ip13 ip13 ip13 ip47 ip33 ip47 ip42 . . . "

4.3. Logistic Regression With N-grams

Since the number of output classes (the same as the number of phrases) is high and the data is limited, we used a two-step classification scheme. In the first stage, we reduce the number of possible classes to 10 using regularized multiclass logistic regression (LR). LR is preferred over support vector machines because it provides class probabilities for each input. The second stage picks the best class among the top 10 classes using a nearest-neighbor search.

We use binary (0/1) features of bigrams of English phonemes for the classifier. Since phoneme sequences for each phrase are short, n-gram models (even bigrams) are very sparse. For a dataset of about 100 phrases and 5 speakers, only 2% of the bigram features have non-zero values. The usage of skip-grams solves this problem to an extent.

Since we have multiple speakers repeating the same phrase, we use cross-speaker bigrams as shown in Fig. 1. The intuition for cross-speaker bigrams is that we expect the phone recognizer to make similar mistakes for future (test) data. Before we count the cross-speaker bigrams, both sequences are aligned to minimize their SED.

The second stage involves a nearest-neighbor search for the test phrase among the top 10 classes picked by LR. Because of this, average rank is only calculated for those utterances found in the top 10, creating artificially low values compared to our other experiments (*). Levenshtein distance is used as the metric for the nearest-neighbor search.

4.3.1. Results

Using the two-stage classification described, accuracies are reported in Table 2. In the case of English, the nearest-

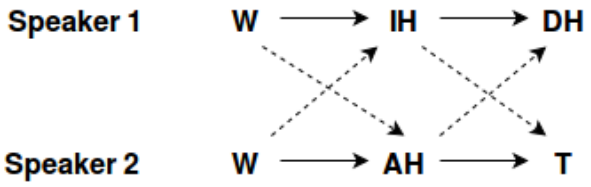

Fig. 1. Cross-speaker bigrams (shown using dotted lines "W IH", "W AH", etc.) illustrated for a short sentence

neighbor search leads to a slight drop in accuracy. For Tamil, which has less data, it improves the performance of logistic regression quite well. However, the algorithm performs poorly on Tamil in comparison to English. This could be either because of the difference in the dataset size or that the phone recognizer (which is trained for English) is not applicable for other languages. We intend to answer this question by collecting more data and using multiple phone recognizers trained on different languages. In addition, we intend to try using high-level features such as utterance length (in phonemes) or duration (in seconds) in this classifier.

Language	% Top 1 Stage I	% Top 1 Stage II	% Top 10	Avg. Rank*
English	71.86	69.75	99.93	1.62
Tamil	34.51	41.6	71.38	2.82

Table 2. Results using logistic regression. Stage I is the output of logistic regression; Stage II is the nearest-neighbor search on the top-10 outputs of logistic regression.

4.4. String Edit Distance

The rest of the experiments consisted of configuring and/or learning paradigms of weights for the SED algorithm. As with the DTW experiment, we looked at 4 metrics per held-out speaker: the number of test phrases that are predicted exactly (% in top 1), the number of test phrases ranked in the top 5 (% in top 5), the number of test phrases ranked in the top 10 (% in top 10), and the *average rank* of the first correct match (whether it be the same phrase spoken by a different speaker or that phrase's model).

First, we used SED without a model, where each test utterance was compared to every other utterance individually. Then, it would pick one of the other utterances with the lowest SED and claim that it was that phrase. The following experiments are variations on this basic SED setup.

4.4.1. Results

As seen in Table 3, the English phoneme experiment significantly outperforms all of the experiments with inferred phones.

Language	Feats	% Top 1	% Top 5	% Top 10	Avg. Rank
English	Eng	**66.748**	**79.820**	**84.232**	**16.721**
English	IP-15	9.804	25.490	36.765	46.180
English	IP-17	11.601	28.350	40.114	47.631
English	IP-30	11.601	28.350	40.114	44.953
English	IP-47	13.154	29.902	41.095	54.262
English	IP-84	12.173	29.739	40.196	61.732
English	IP-92	14.134	29.330	39.624	64.761
English	IP-101	11.111	28.922	38.235	67.253
English	IP-123	11.029	28.431	39.788	69.479
English	IP-171	10.621	27.206	37.173	71.710

Table 3. Results for the standard string edit distance experiments

4.5. Gaussian Model

As in the DTW experiment, utterances of the same phrase across different speakers have their SED scores averaged in order to create a model. In testing, each phrase is compared to each model by its z-score, and all scores are sorted.

4.5.1. Results

The results (Table 4) drop sharply from using the English phonemes to using the IPs when experimenting on English. This drop is even sharper when we only look at 3 of the 12 English speakers. On the other hand, the best of the results for Tamil are using the IPs, and these values are over 20% lower than the best English results.

4.6. Weighted String Edit Distance

The following experiments share the same methodology as the Gaussian model described above, with the exception of using different weights for edge costs in calculating the SED.

4.6.1. Phonetic Distance Weights (PD)

Using the vowel and consonant charts of the International Phonetic Alphabet as a guide, a table was created to establish SEDs between all possible pairs of phones. These distances, ranging from 0 (identical) to 1 (as dissimilar as possible), were then used as the weights in calculating the SED.

These experiments were only performed on the English phonemes, since there is no way to find the phonetic distance between the IPs. As seen in Table 5, English still performs the best, but Tamil does better than the smaller set of English, showing that the number of speakers matters. All of the results are worse than the results from the standard SED experiments.

4.6.2. Articulatory Feature Weights (AF)

In order to create the IP trees for each language, it was necessary to calculate values for the articulatory features of each English phoneme using the given data. In these experiments,

Language	Feats	% Top 1	% Top 5	% Top 10	Avg. Rank
English	Eng	**87.173**	98.856	99.265	1.192
English	IP-15	80.310	99.265	99.265	1.238
English	IP-17	76.389	99.265	99.265	1.307
English	IP-30	80.474	99.183	99.265	1.262
English	IP-47	77.451	98.693	99.265	1.314
English	IP-84	79.248	98.856	99.265	1.316
English	IP-92	78.023	99.020	99.265	1.314
English	IP-101	79.575	98.611	99.265	1.312
English	IP-123	75.654	98.693	99.183	1.356
English	IP-171	76.716	98.693	99.265	1.347
Tamil	Eng	59.216	92.941	99.020	2.409
Tamil	IP-14	60.392	98.627	99.608	1.821
Tamil	IP-17	57.451	97.843	99.608	1.925
Tamil	IP-24	59.608	97.647	99.804	1.749
Tamil	IP-51	62.745	97.255	99.608	1.846
Tamil	IP-82	62.157	97.255	99.608	1.755
Tamil	IP-93	60.980	97.255	99.608	1.795
Tamil	IP-103	58.824	98.039	99.608	1.839
Tamil	IP-118	60.392	97.255	99.804	1.745
Tamil	IP-165	**62.941**	95.882	99.412	1.820
Eng-Small	Eng	**80.392**	87.255	88.235	7.072
Eng-Small	IP-15	42.484	89.542	93.791	5.115
Eng-Small	IP-17	40.523	86.275	92.484	6.246
Eng-Small	IP-30	45.752	86.275	94.771	4.799
Eng-Small	IP-47	39.216	84.967	94.118	5.151
Eng-Small	IP-84	38.235	82.026	90.850	6.634
Eng-Small	IP-92	37.908	79.412	88.889	6.760
Eng-Small	IP-101	36.928	81.373	91.503	6.651
Eng-Small	IP-123	35.294	74.837	85.948	8.643
Eng-Small	IP-171	39.869	78.431	89.216	7.506

Table 4. Results for Gaussian experiments

Language	% Top 1	% Top 5	% Top 10	Avg. Rank
English	59.722	97.386	99.020	1.724
Tamil	39.020	88.431	97.843	2.845
Eng-Small	21.242	67.974	78.758	14.620

Table 5. Results for phonetic distance weights in SED using English phonemes

we took the Euclidean distances of these values to use as weights in calculating the SED.

These experiments were only performed on the English phonemes since the articulatory features are only available for known phones. The results, seen in Table 6, are similar to the phonetic distance weight results.

Language	% Top 1	% Top 5	% Top 10	Avg. Rank
English	63.562	99.183	99.265	1.525
Tamil	37.843	94.510	99.216	2.714
Eng-Small	26.471	72.876	82.680	12.368

Table 6. Results for articulatory feature weights in SED using English phonemes

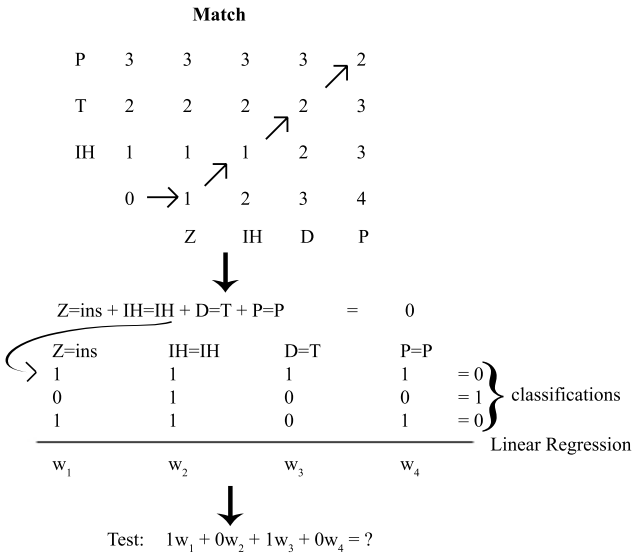

Fig. 2. A toy example illustrating the weight learning process

4.7. Learned Weights

We began this process with the original 0 (match) / 1 (non-match) weights for SED. Using backpointers, we find the path of the minimal SED. With the reference and the hypothesis aligned, we get our weight labels from each pair along the path. Once all of the paths are found, the weights are combined into a giant vector of possible weights, such that if one instance has that weight, it is set to 1; and otherwise is set to 0.

These strings of values are summed, and set to either 1 or 0. If the two phrases being compared match (speakers cannot be compared to themselves), then the phrase is set to 0; and otherwise is set to 1. We then use linear regression to find the new weights, using them to predict the value of the held-out set of phrases (all from one held-out speaker), such that it predicts which phrase is the best match. The weights are trained so that the smaller the score, the better the match between two phrases. This way we can use the new weights in the next iteration for our SED costs and still continue to find the cheapest path (Fig. 2).

This results in a sparse vector where each of the listed pairs equals 1, and the other pairs that were found in other utterance comparisons equal 0. Each sparse vector is set to either 0 or 1, and we learn the weights per phrase, per held-out speaker. Once the weights are found through linear regression, they are used as the new costs in the SED algorithm, and the rest of the procedure continues for *n* iterations.

Here we used 3 speakers in our English collection, instead of all 12, in order to find the best use of this process. This way, we could judge how many iterations would be optimal without wasting computational time and resources.

4.7.1. Results

Using the smaller English dataset, we see in Table 7 that using one iteration of the weight learning outperforms the standard SED Gaussian results.

Iterations	% Top 1	% Top 5	% Top 10	Avg. Rank
No Weights	80.392	87.255	88.235	7.072
1x	**88.235**	**95.425**	**96.732**	**1.913**
2x	79.739	90.196	92.484	2.837
3x	76.797	87.255	92.157	3.433
4x	79.412	90.850	93.137	3.047
5x	76.797	89.216	92.484	3.700
6x	81.046	91.176	93.464	3.187

Table 7. Results for learned weights experiments using the "Eng-Small" dataset and English phonemes

Results obtained from repeating the one-iteration process on the IPs from the Eng-Small dataset and all of the Tamil dataset are shown in Table 8. The English phones work much better than IPs for Eng-Small. The English phones and the IP-103 experiment are comparable for Tamil, though both are less effective than the Gaussian experiment. Some of the experiments with high numbers of IPs were omitted due to computational expense.

Language	Feats	% Top 1	% Top 5	% Top 10	Avg. Rank
Eng-Small	Eng	**88.235**	**95.425**	**96.732**	**1.913**
Eng-Small	IP-15	13.072	34.641	48.039	27.437
Eng-Small	IP-17	15.033	40.523	55.556	22.447
Eng-Small	IP-30	21.895	49.020	60.458	16.623
Eng-Small	IP-47	16.667	47.712	58.824	19.823
Eng-Small	IP-84	18.627	42.157	55.556	20.077
Eng-Small	IP-92	19.608	41.503	56.863	18.210
Eng-Small	IP-101	15.033	37.908	49.673	20.467
Tamil	Eng	49.608	77.255	86.078	8.128
Tamil	IP-14	25.098	53.529	65.882	13.736
Tamil	IP-17	33.529	65.098	76.471	10.532
Tamil	IP-24	37.843	70.784	83.137	6.375
Tamil	IP-51	47.647	75.686	86.275	5.816
Tamil	IP-82	47.647	74.510	82.745	7.000
Tamil	IP-93	46.275	76.471	84.510	6.569
Tamil	IP-103	49.608	76.667	84.314	6.413

Table 8. Results for SED with learned weights, one iteration

5. DISCUSSION AND FUTURE WORK

Throughout all of our experiments, we have seen that English phonemes work the best with English. Specifically, SED weight learning through linear regression appears best.

However, the results are less clear for Tamil. It seems that for Tamil, the IPs work somewhat better than the English phonemes. However, unlike for the English 3-speaker experiments, learning SED weights does not appear to improve upon results using the default (Levenshtein distance) weights. This may be due to overfitting, especially considering that the English 3-speaker experiments do poorly with the IPs as well.

From this we conclude that the more speakers we can acquire, the better the result will be. This is to be expected. From here, the question arises: How many speakers are needed to build an adequate system, and which ones are most useful? We have begun to investigate how to determine the quality of utterances, so that the program knows when a user should repeat what they said. We have also started using an empirical method to determine whether a speaker is of low quality or utility overall and therefore should be left out of training. This approach also enables us to identify the speakers that are most useful for classification, so that we use only as many such speakers as are necessary to improve the model, in order to optimize for both accuracy and runtime. Preliminary results on the full English speaker set suggest that for that model, a set of five specific speakers is optimal; we expect this will vary based on the language and the speakers.

We are currently collecting Telugu data, but have not acquired sufficient data to reliably test these techniques on that language yet. We are also collecting utterances for Mandarin Chinese and will use that as our next language to experiment with, since we have access to numerous speakers.

We will finish developing the smartphone application by incorporating our best classification algorithm in order to predict phrases in real time. We will then test the application at both CMU and UPMC.

At the moment, our 102 phrases are fairly phonetically distinct, but our full set of 750-plus phrases will be harder to distinguish. It remains to be seen how well our classification algorithms will perform with several times as many phrases. This is a major open question going forward. Therefore, it will be necessary to experiment with subsentential approaches such as low-resource keyword spotting, which will enable the system to recognize important patterns in the phonemes/IPs and limit the search set.

With data collected from a dialogue system we have developed that uses only our predefined phrases, we plan on using dialogue-state tracking to improve performance of the recognition. For example, one would be less likely to expect a user to say "hello" again halfway through the conversation. Combining this model with the phonetically-based model should improve classification.

Finally, because no one modeling approach has yet been conclusively shown to be best overall, in moving toward a practical system that will run on the app in a reasonable amount of time, we will focus more attention on the English phones representation than the IPs. We intend to try combining the Gaussian approach and the weight-learning approach that uses linear regression. We also continue to find and apply other algorithmic optimizations; for instance, using a beam search for the edit distance grid calculations improves both speed and accuracy.

6. REFERENCES

[1] Andreas Kathol, Kristin Precoda, Dimitra Vergyri, Wen Wang, and Susanne Riehemann, "Speech translation for low-resource languages: The case of pashto," in *INTERSPEECH*, 2005, pp. 2273–2276.

[2] David Stallard, Fred Choi, Jacob Devlin, Kriste Krstovski, Prem Natarajan, Ralf Meermeier, Rohit Prasad, Shankar Ananthakrishnan, and Shirin Saleem, *The BBN Transtalk Speech-to-Speech Translation System*, INTECH Open Access Publisher, 2011.

[3] Ace Sarich, "Phraselator, one-way speech translation system," 2001.

[4] Sebastian Stüker and Alex Waibel, "Towards human translations guided language discovery for asr systems," in *SLTU*, 2008, pp. 76–79.

[5] Robert S Melvin, Win May, Shrikanth Narayanan, Panayiotis G Georgiou, and Shadi Ganjavi, "Creation of a doctor-patient dialogue corpus using standardized patients," in *Language Resources and Evaluation Conference*, 2004.

[6] Sunayana Sitaram, Shrikant Palkar, Yun-Nung Chen, Alok Parlikar, and Alan W Black, "Bootstrapping text-to-speech for speech processing in languages without an orthography," *Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on Acoustics*, pp. 7992–7996, 2013.

[7] Prasanna Kumar Muthukumar and Alan W Black, "Automatic discovery of a phonetic inventory for unwritten languages for statistical speech synthesis," *Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on Acoustics*, pp. 2594–2598, 2014.