SENTENCE RECOGNITION FROM ARTICULATORY MOVEMENTS FOR SILENT SPEECH INTERFACES

Jun Wang[1, 2, 3], Ashok Samal[1], Jordan R. Green[2, 3], Frank Rudzicz[4]

[1]Department of Computer Science & Engineering
[2]Department of Special Education & Communication Disorders
University of Nebraska – Lincoln, Lincoln, NE, United States
[3]Munroe-Meyer Institute, University of Nebraska Medical Center, Omaha, NE, United States
[4]Department of Computer Science, University of Toronto, Toronto, ON, Canada
{junwang, samal}@cse.unl.edu, jgreen4@unl.edu, frank@ai.toronto.edu

ABSTRACT

Recent research has demonstrated the potential of using an articulation-based silent speech interface for command-and-control systems. Such an interface converts articulation to words that can then drive a text-to-speech synthesizer. In this paper, we have proposed a novel near-time algorithm to recognize whole-sentences from continuous tongue and lip movements. Our goal is to assist persons who are aphonic or have a severe motor speech impairment to produce functional speech using their tongue and lips. Our algorithm was tested using a functional sentence data set collected from ten speakers (3012 utterances). The average accuracy was 94.89% with an average latency of 3.11 seconds for each sentence prediction. The results indicate the effectiveness of our approach and its potential for building a real-time articulation-based silent speech interface for clinical applications.

Index Terms— Sentence recognition, silent speech interface, support vector machine, laryngectomy

1. INTRODUCTION

Oral communication plays an important role in social life. Persons with speech impairments (caused, e.g., by laryngectomy, which is partial or complete surgical removal of larynx) struggle with their daily communication [1]. Each year, about 15,000 new cases of laryngeal and hyperlaryngeal cancer are diagnosed in the United States [2] and there are an estimated 16,500 tracheo-oesophageal surgeries every year in the UK [3]. However, currently, there are only limited treatment options for those individuals, which either produces an un-natural voice (i.e., by electrolarynx) or is limited by slow manual input (i.e., as in typing-based Augmentative and Alternative Communication devices, AAC) [1]. New assistive technologies are needed to provide a more efficient and natural mode of oral communication for these individuals.

Silent speech interfaces (SSIs), although still experimental [4], may provide an efficient communication modality. Articulation-based SSIs convert silently produced articulatory movement or vocal tract data into orthographic transcriptions that can be used to drive a text-to-speech synthesizer (TTS) or to trigger the playback of pre-recorded sounds. An advantage of using pre-recorded sounds is that the individual's own voice can be recorded and replayed post laryngectomy [2, 3, 4].

Two major challenges of developing SSIs are the lack of portable and fast data acquisition devices (hardware) and of sufficient algorithms (software) to convert non-acoustic data to speech text. Electromagnetic articulography (EMA) is a promising development towards better hardware [4]. Fagan et al. have shown the potential of their EMA-based silent speech interface for command-and-control applications by successfully classifying a set of words from movements of sensors affixed to the tongue and lips [3, 5]. Our study is focused on the development of a fast and accurate algorithm that converts articulation to text.

Articulatory data can improve the accuracy of automatic word recognition for the voiced speech of both healthy [6, 7] and neurologically impaired individuals [8]. This typically involves the use of *articulatory features* (AFs) which include lip rounding, tongue tip position, and manner of production, for example. Phoneme-level AF-based approaches often obtain word recognition accuracies less than 50% [6] because articulation can vary significantly within those categorical features depending on the surrounding sounds and the speaking context [9]. These challenges motivate a higher-level unit of recognition.

Sentence-level recognition has rarely been investigated due in part to the difficulty in training appropriate models. Our long-term goal is to recognize a set of functional sentences (i.e., those used by AAC users in practice) that drive EMA-based silent speech interfaces for clinical applications. This paper presents a novel sentence-level and near-time recognition algorithm. The algorithm was tested using a functional sentence dataset, which is part of our ongoing data collection. The algorithm is characterized by the following features: (1) recognition is sentence-level, rather than phoneme-level; (2) it is based on continuous articulatory movements, rather than on discrete AFs; (3) it uses a dynamic thresholding technique based on probability change patterns; and (4) it is extensible, which means a variety of classifiers can be built-in easily. The algorithm will provide the recognition component of our future articulation-based SSI.

2. DESIGN & METHOD

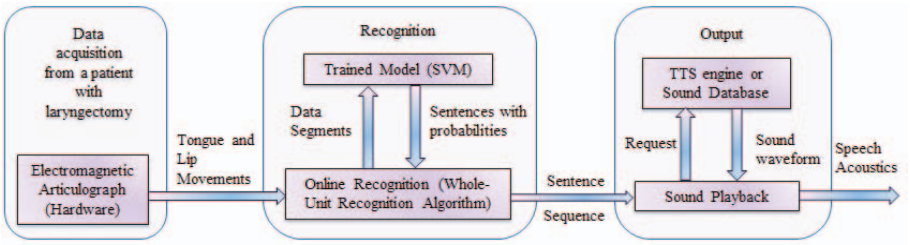

Fig. 1. Design of the EMA-based silent speech interface

Fig. 1 illustrates the design of our EMA-based SSI, which contains three components: (a) data acquisition, (b) online (sentence) recognition, and (c) sound playback or synthesis. This paper focuses on the online recognition component whose goal is to recognize a set of functional sentences, *S*. The central recognition problem is to convert a time-series of spatial configurations of multiple articulators to time-delimited sentences. Here, a spatial configuration is an ordered set of 3D locations of the articulators. To attain real-time performance, we combine the segmentation and identification into a variable size moving window algorithm. The algorithm is based on the premise that a sentence will have its highest matching probability given an observation window with an appropriate width and starting point. A trained classifier that derives these matching probabilities is embedded into the algorithm, as described in the following section.

2.1. Classifier Training

A support vector machine (SVM) [10] was trained using pre-segmented articulatory movement time-series data from multiple sensors associated with known sentences (training data). SVMs are widely used classifiers that find a separating hyperplane between classes by maximizing the margin between them. A kernel function is used to describe the distance between two data points (i.e., *u* and *v* in Eq. 1). The following radial basis function (RBF) was used as the kernel function in this study, where λ is an empirical parameter:

$$K_{RBF}(u,v) = \exp(1 - \lambda \,||\, u - v \,||) \qquad (1)$$

The time required to train this model is not relevant here because the training component is developed off-line, i.e., before the SSI is deployed in online recognition. Only the time to predict sentences is important in real-time applications. To obtain a high speed in prediction, a *direct mapping* strategy was used, in which the input data was minimally processed before being fed into the SVM (directly mapped to words). Here, the motion paths of sensors attached to the tongue and lips are segmented for each sentence and time-normalized to a fixed-width (SVMs typically require samples to have a fixed length of samples) and concatenated as a single vector of attributes, which forms a sentence observation. Furthermore, to compare the accuracy of our SVM technique to a more common time-series classification approach, we also tested the classification using dynamic time warping (DTW) [3].

2.2. Online Recognition

The trained classifier is used to recognize sentences from continuous (unsegmented) tongue and lip movement data. Here, a prediction window with variable boundaries traverses the sequence of tongue and lip movement data to recognize sentences and their locations within the window based on the probabilities returned by LIBSVM, which extends the generic SVM by providing probability estimates transformed from SVM decision values [11].

2.2.1. Parameters

Several parameters were obtained from a training data set of sentences *S* (|*S*|=*N*) before being used in online recognition:

- l_{max}: maximum length of sentences
- l_{min}: minimum length of sentences
- $th_c[N]$: an array of candidate thresholds, one for each sentence in *S* that represents the minimum probability for a sample to be considered as a candidate.
- $th_d[N]$: an array of decision thresholds, one for each sentence in *S* which specifies the probability necessary to verify a candidate. These are the mean identification probabilities across training sequences for each sentence.
- *W*: span of the prediction window, specified by its left and right boundaries, [W_l, W_r].

The algorithm also uses the following adjustable empirical parameters:

- Δt: step size with which the window *W* moves forward.
- Δlen: step size with which the size of estimated sentence length is incremented in the process of generating probabilities.

2.2.2. Recognition Algorithm

Fig. 2 gives the details of the proposed algorithm. First, at each time *t*, possible candidates are generated by the sub-function GenerateCandidates. The probabilities of all possible lengths for each sentence at time *t* were explored within multiple time spans between ($t + l_{min}$) and ($t + l_{max}$) in steps of Δlen. The duration of sentences ranged from 0.50 (l_{min}) to 1.19 (l_{max}) seconds in this dataset. The window *W* then moves by Δt and the process is repeated. The offset of the probability function varied considerably across sentences, which makes it difficult to identify a sensitive candidate threshold. Therefore, the probabilities associated with each sequence are baseline-corrected by subtracting the average probability derived from the first l_{min} seconds of the test sequence. Furthermore, each sentence has its own thresholds (th_c and th_d). At each *t*, the highest probability across different lengths is retuned as the probability at *t* for a sentence *s*. If the probability is greater than the candidate threshold (th_c), a candidate is saved in *C*. In the early stage of this work, th_c (= 0.083 for all sentences) and th_d (=0.6) were given based on training probabilities and observation.

All candidates are then verified according to their probabilities (Lines 4-19). For a sentence *s*, there are two possibilities in terms of number of candidates. First, if there is only one candidate *c* (represented by a tuple <*s*,*t*>) for *s*, then if its

Whole-Unit Recognition Algorithm
Input: *sequence*
Parameters: S, l_{min}, l_{max}, th_c, th_d, Δt, Δlen,

```
1   t = 0;    W_l = 0; W_r = 0; C = ø; R = ø;
2   while (t < |sequence|- l_max)
3     GenerateCandidates(sequence, t, W_l, W_r, C);
4     if !empty(C )   // if found candidates in window [W_l, W_r]
5         W_r = t;  //adjust right boundary of prediction window
6         for ∀ s ∈ S
7             if there is one candidate, c, s.t. prob(c) ≥ th_d(s)
8                 R = R ∪ {c};  // c=<s, t>, s is recognized at t
9                 W_l = c.t;  //adjust left boundary of the window W
10            else if ∃ c_1, c_0, s.t. t_0 < t_1, prob(c_1) < prob(c_0)
11                R = R ∪ {c_0};  // s is recognized at t_0
12                W_l = t_0;  // c_0 = <s, t_0>;
13            end
14        end
15        R = CheckTimeLocationConstraint(R);
16        Output(R);   // output all c's in R in chorological order
17        C = C – R;   // remove all c's in R from C
18        R = ø;    // clear R;
19    end  // concludes if in line 4
20    t = t + Δt;     //keep reading data
21  end  // concludes while in line 2
```

SubFunc GenerateCandidates($sequence$, t, W_l, W_r,C)
Parameters: l_{min}, l_{max}, Δlen

```
1     for ∀ s ∈ S
2         for (i = l_min; i<l_max; i = i + Δlen)
3             probs(s, t + i) = GetProbs(sequence, t, t + i);
4     end
5     for ∀ s∈ S, prob(s, t) = GetMax(probs, W_l, W_r);  end
6     for ∀ s ∈ S, RemoveBaseline(prob); end
7     for ∀ s ∈ S
8         if prob(s, t)  ≥ th_c(s)
9                 c = <s, t>;   C = C ∪ c;   end
10    end // C is the candidate list
EndSubFunc
```

Fig. 2. Whole-unit online recognition algorithm.

probability is greater than the decision probability $th_d(s)$, the sentence is recognized with that hypothesis. Alternatively, there could be more than one candidate for s. Here, the trend in the change of probability is analyzed within the window W. If the probabilities for s are decreasing, implying ongoing decreases, the candidate for s is confirmed (Line 10-11 in Fig. 2); otherwise, the decision-making is delayed. We assume that the probabilities for s cannot increase indefinitely as t moves forward.

The Time Location Constraint (TLC) (Line 15 in Fig. 2) is as follows: if the difference between times t_1 and t_2 is less than l_{min}, they are considered as a single time location. Only the sentence with the highest probability is retained at one time location.

The time taken by the algorithm is in $O(n \times l \times |S|)$, where n is the length of the input sequence in time, l (i.e., $(l_{max} - l_{min}) / \Delta len$), the number of estimated length and $|S|$ (number of sentences) can be treated as a constant for a given dataset for evaluation.

Two measures, *prediction location offset* (machine-independent), and *prediction processing time* (*latency)* (machine-dependent), were used to evaluate the efficiency of the algorithm. *Prediction location offset* is defined as the difference in location (indicated using time) between where a sentence was spoken and the time it was recognized. It gives a rough estimate of how much information is needed for predicting a sentence. *Latency* is the actual CPU time for a sentence prediction (prediction time minus the sentence onset time)

3. DATA COLLECTION & PROCESSING

3.1. Participant, stimuli, and procedure

Ten healthy American English female speakers participated in the data collection. Each speaker participated in one session in which they repeated multiple iterations of a sequence of sentences. Twelve sentences for basic greeting and conversation were selected form a list of most frequently used sentences among AAC users [12]. In total, 3012 sentences (in 251 sequences) were obtained and used in this experiment.

The electromagnetic articulograph AG500 (Carstens Inc. Germany) was used to register the 3-D movements of the tongue, lip, and jaw when a subject was talking. The AG500 records movements by establishing a calibrated electromagnetic field in a cube that can track the movements of tiny sensor coils that were attached on the surface of the tongue, lips, and jaw using dental glue. The spatial precision of motion tracking using the AG500 is approximately 0.5 mm [13].

Fig. 3 shows the positions of the 12 sensors. Three head sensors, HC (Head Center), HL (Head Left) and HR (Head Right) were collected to perform head-orientation normalization. Data from four tongue sensors named T1 (Tongue Tip), T2 (Tongue Blade), T3 (Tongue Body Front), and T4 (Tongue Body Back), and two lip sensors (Upper and Lower Lip) were used for analysis. The movements of three jaw sensors, JL (Jaw Left), JR (Jaw Right), and JC (Jaw Center), were recorded for future use.

3.2. Data processing

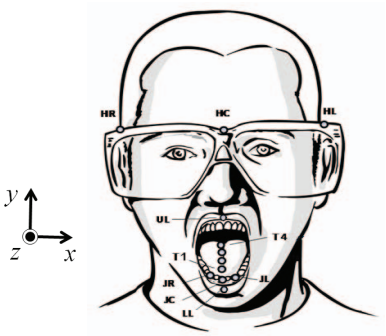

Fig. 3. Positions of sensors attached on the subject's head, face

Prior to analysis, the time-series data of sensor locations recorded using EMA went through a sequence of preprocessing steps. First, head movements were subtracted from the tongue and lip locations. The orientation of the derived 3-D Cartesian system is displayed in Figure 3. Second, a zero-phase lag low-pass filter (at 10 *Hz*) was applied for removing signal noise. Third, all sequences were manually segmented and annotated with sentences. Only y and z coordinates of the sensors (i.e., T1, T2, T3, T4, UL, and LL) were used for analysis because the movement along the side-to-side axis is not significant in normal speech production.

4. RESULTS & DISCUSSION

Leave-one-out cross validation was conducted within each subject. The average classification (training) accuracy was 93.76% (std. dev. σ=2.39) across subjects using our approach (direct mapping and SVM). Although the DTW approach obtained a higher accuracy of 98.09% (σ=1.55), our SVM approach (which took an average only of 7.98 ms for a single sentence classification) was significantly ($p<0.00001$) faster than DTW approach (which took 3069.2 ms on average). Thus, only the SVM approach was used in the online recognition experiment, where a sentence was recognized correctly if the predicted time was less than 100 ms before or after the actual onset time.

On average, the online recognition *accuracy* across subjects was 94.89% (std. dev. σ=3.72). The average *prediction location offset* and *latency* was 0.23 (σ=0.04) and 3.11 (σ=0.97) seconds for each sentence, respectively. The high accuracy shows the effectiveness of our proposed algorithm. The short *delay* indicated that our approach was able to make a prediction based on even a small amount of information, making it feasible for real-time applications. A short *latency* gives an estimate of how much CPU time is needed for each sentence prediction (we used a PC with a 2.5 GHz dual core CPU and 6GB RAM). The low standard deviations of these measures across subjects indicate that the algorithm is effective for multiple persons. Fig. 4 shows an example of the probabilities on a selected sequence where peaks occur in the presence of known sentences.

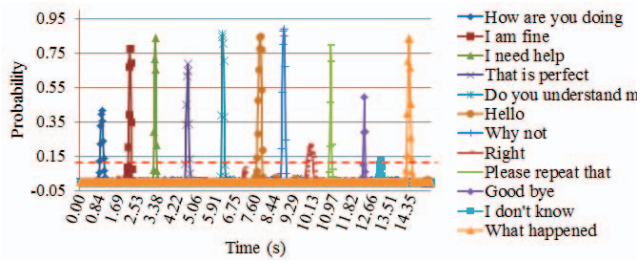

Fig. 4. Probability (baseline removed) of sentence candidates generated by the online recognition algorithm at time points on a test sequence. The red dashed line is the candidate threshold (th_c).

5. CONCLUSION & FUTURE WORK

Experimental results indicate the effectiveness and efficiency of our proposed whole-sentence recognition algorithm from articulatory movements and its potential for building a real-time articulation-based SSI, which can be used by non-vocal individuals.

Although the current results are encouraging, the online algorithm still has room for improvement. First, automated methods are needed to find optimal parameter settings during training. Second, decisions during online prediction could be improved using more sophisticated criteria than the peak value in the probability function. Third, faster DTW algorithms [14] and other classifiers (e.g., HMM [15]) will also be investigated.

6. ACKNOWLEDGEMENTS

This work was in part funded by a grant awarded by the National Institutes of Health (R01 DC009890/DC/NIDCD NIH HHS/United States) and the Barkley Trust, University of Nebraska - Lincoln. We would like to thank Dr. Tom D. Carrell, Dr. Lori Synhorst, Cynthia Didion, Rebecca Hoesing, Kate Lippincott, Kayanne Hamling, and Kelly Veys for their contribution to subject recruitment, data collection, and data processing.

7. REFERENCES

[1] B. J. Bailey, J. T. Johnson, and S. D. Newlands, *Head and neck surgery - otolaryngology*, Lippincot, Williams & Silkins, Philadelphia, PA, 4th Edition, pp. 1779-1780, 2006.

[2] J. Wang, J. R. Green, A., Samal, and T. Carrell, "Continuous vowel recognition from articulatory movements," *IEEE Intl. Conf. on Signal Processing & Communication Systems*, 2010.

[3] M. J. Fagan, S. R. Ell, J. M. Gilbert, E. Sarrazin, and P. M. Chapman, "Development of a (silent) speech recognition system for patients following laryngectomy," *Medical Engineering & Physics,* vol. 30, no. 4, pp. 419–425, 2008.

[4] B. Denby, T. Schultz, K. Honda, T. Hueber, J. M. Gilbert, J. S. Brumberg, "Silent speech interfaces," *Speech Communication* vol. 52, pp. 270-287, 2010.

[5] J. M. Gilbert, S. I. Rybchenko, R. Hofe, S. R. Ell, M. J. Fagan, R. K. Moore, and P. Green, "Isolated word recognition of silent speech using magnetic implants and sensors," *Medical Engineering & Physics*, vol. 32, no. 10, pp. 1189-1197, 2011.

[6] K. Livescu, O. Cetin, M. Hasegawa-Johnson, S. King, C. Bartels, N. Borges, A. Kantor, P. Lal, L. Yung, A. Bezman, S. Dawson-Haggerty, B. Woods, J. Frankel, M. Magimai-Doss, and K. Saenko, "Articulatory feature-based methods for acoustic and audio-visual speech recognition: Summary from the 2006 JHU Summer Workshop," *Proc. ICASSP*, 2007.

[7] S. King, J. Frankel, K. Livescu, E. McDermott, K. Richmond, and M. Wester, "Speech production knowledge in automatic speech recognition," *Journal of the Acoustical Society of America*, vol. 121, no. 2, pp. 723-742, 2007.

[8] F. Rudzicz, "Articulatory knowledge in the recognition of dysarthric speech," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 4, pp. 947-960, 2011.

[9] E. Uraga, and T. Hain. "Automatic speech recognition experiments with articulatory data," *Proc. Interspeech*, 2006.

[10] C. Cortes and V. Vapnik. "Support-vector networks," *Machine Learning*, vol. 20, pp. 273-297, 1995.

[11] C.-C. Chang and C.-J. Lin. "LIBSVM: a library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2(3), no. 27, pp.1-27, 2011.

[12] D. R. Beukelman, K. M., Yorkston, M. Poblete, and C. Naranjo, "Analysis of communication samples produced by adult communication aid users," *Journal of Speech and Hearing Disorders*, vol. 49, pp. 360-367, 1984.

[13] Y. Yunusova, J. R. Green, and A. Mefferd, "Accuracy assessment for AG500 electromagnetic articulograph," *Journal of Speech, Language, and Hearing Research*, vol. 52, no. 2, pp. 547-555, 2009.

[14] S. Salvador and P. Chan, "FastDTW: Toward accurate dynamic time warping in linear time and space," *3rd Workshop on Mining Temporal and Sequential Data*, *ACM KDD*, 2004.

[15] P. Heracleous, and N. Hagita, "Automatic recognition of speech without any audio information," *Proc. ICASSP*, pp. 2392-2395, 2011.