HMM, recognition, speech, disorders

Marek WIŚNIEWSKI[1], Wiesława KUNISZYK-JÓŹKOWIAK

AUTOMATIC DETECTION AND CLASSIFICATION OF PHONEME REPETITIONS USING HTK TOOLKIT

The therapy of stuttering people is based on a proper selection of texts and then on a practice of their articulation by reading or narration. The texts are chosen on the basis of kind and intensity of dysfluencies appearing in a speech. Thus there is still a requirement to find effective and objective methods of analysis of dysfluent speech. Hidden Markov models are stochastic models widely used in recognition of any patterns appearing in a signal. In the work a simple monophone system based on the Hidden Markov Model Toolkit was built and tested in the context of detection and classification of phoneme repetitions – a common speech disorder in the Polish language.

1. INTRODUCTION

Automatic stuttering recognition systems can provide an objective and consistent measure of diagnosis and therapy processes of stuttering people. It automates counting and classification of speech disturbances helping therapists in a better selection of a treatment. The main aim of studies on such systems focuses on an improvement of judgment quality of speech fluency as well as versatility and simplicity.

The methods of automatic detection and classification of speech disorders are usually based on algorithms of acoustic analysis, neural networks or statistical models.

The acoustic analysis is based on experimental determination of specific distinctive features and development of a suitable algorithm that gives possibility to find a given disorder in an acoustic signal [2]. For example, in order to detect phoneme prolongations, a duration time is used and for detection of phoneme repetitions the spectrum correlation algorithms can be used. Additionally, the classification based on the kind of disturbed phoneme is desirable (nasal or whispered phonemes, vowels, consonants, etc). Another issue is that the algorithms are reliable for one speaker and unreliable for another one. Therefore every kind of disorder requires a separate algorithm to be developed and also adopted to each individual speaker.

An interesting fact is that the audio features can be complemented by visual cues in the stuttering analysis. These can be measures of lips, jaw or tongue movement [1, 5]. For example, the jaw movement (displacement range, duration time) correlates well with the speech motor deficiencies and stuttering level [1].

Artificial Neural Networks (ANN) are mathematical models used in analysis of both a fluent and dysfluent speech [5, 8, 9, 10, 13]. ANN stuttered speech analyzers work as follows. Training samples are parameterized (autocorrelation, spectral information, envelope, etc.) and marked as fluent or dysfluent (prolonged, repeated or others) . Next they are given to the network input and then it is adjusted automatically to achieve the proper output. For each type of disturbances, separate or integrated ANN of a different type (Kohonen, MLP) can be trained. After training a tested utterance can be placed to the input and the network classifies it by a proper output response.

The last of the common approaches are statistical methods. They usually utilize hidden Markov models (HMM) commonly used in speech recognition systems [3]. HMMs can be used as simple pattern recognition systems, in which each model is prepared to detect a separate dysfluence [11], or in a more advanced way, they are used only to obtain a phonetic transcription [12]. In that case for each phoneme in a language, a HMM model is prepared and dysfuencies are detected by analysis of transcribed utterances.

Each described approach has advantages and disadvantages. Under specific circumstances they could give good results (above 80% of the detected dysfluencies). But HMM based methods seem to be

[1] email: marek.wisniewski@poczta.umcs.lublin.pl.

the most universal and simplest. They give the possibility to detect and classify dysfluencies as well as enable identification of disturbed phonemes which is very important with respect to the therapy process.

2. THE RECOGITION SYSTEM

In the work the Hidden Markov Model Toolkit (HTK) [6] was utilized. It is one of the best known systems designed for HMMs manipulation and it was primarily designed for speech recognition tasks.

The idea was to build and test the system that generates phonetic transcription from input utterances and then finds and identifies phoneme repetitons. The work included preparation of training materials, selection of a suitable HMM model structure, model training process and testing.

2.1. TRAINING MATERIALS

As training materials, recordings from our laboratory were used. For training purposes there were selected 425 utterances coming from one person (each utterance lasting from about 100 ms to several seconds). For each of them phonetic transcription was written. There were 39 symbols used: 37 for phonemes and 2 for silence [7].

For a transcription purpose the own set of phoneme symbols was prepared and used (Table 1). The set number one includes Polish characters and was used in filenames. It facilitates transcription writing and management of a training set. The set number two includes only ASCII characters and was used in HTK. Transcriptions from filenames were converted to the latter set using PERL script and placed in the MLF file (Master Label File - the form of transcription file used in HTK).

The proprietary symbol sets were prepared because the other sets give much worse human-readable transcriptions.

Table 1. The phoneme symbol sets used in the work and their comparison to the other symbol sets.

	Slavistic alphabet	IPA	Proprietary symbol set 1 (used in filenames)	Proprietary symbol set 2 (used in HTK)	X-Sampa
1.	a	a	a	a	a
2.	b	b	b	b	b
3.	c	t͡s	c	c	ts)
4.	č	t͡ʃ	cz	cz	tS)
5.	ć	t͡ɕ	ć	ci	ts\)
6.	d	d	d	d	d
7.	ʒ	d͡z	dz	dz	dz)
8.	ʑ	d͡ʑ	dź	dzi	dz\)
9.	e	ɛ	e	e	e
10.	ǯ	d͡ʒ	dż	drz	dZ)
11.	ę	ɛ̃	ę	e~	E~
12.	f	f	f	f	f
13.	g	g	g	g	g
14.	ǵ	ɟ	g	gi	g\
15.	X	x	h	h	x
16.	i	i	i	i	i
17.	i̯	j	j	j	j
18.	k′	c	k	k	k
19.	l	l	l	l	l
20.	u̯	w	ł	l~	w
21.	m	m	m	m	m
22.	n	n	n	n	m
23.	ń	ɲ	ń	ni	n_j
24.	o	ɔ	o	o	o
25.	ǫ	ɔ̃	ą	a~	o~
26.	p	p	p	p	p
27.	r	r	r	r	r

28.	s	s	s	s	s
29.	š	ʃ	sz	sz	S
30.	ś	ɕ	ś	si	s\
31.	t	t	t	t	t
32.	u	u	u	u	u
33.	v	v	w	w	v
34.	y	ɨ	y	y	1
35.	z	z	z	z	z
36.	ź	ʑ	ź	zi	z\
37.	ž	ʒ	ż	rz	Z

2.2. AUDIO PARAMETRIZATION

Parameters of audio recordings were as follows: sample frequency 22050Hz and amplitude resolution 16 bits.

Signal feature vectors consisted of 39 elements: 12 MFCC, 1 energy parameter and the first and second derivatives (delta parameters).

Signals were filtered by the pre-emphasis filter and divided into frames of 512 samples length (~23 ms). Each frame was shifted about 50% of sample length (~11.5 ms) compared to the previous one.

The further process of parameterization was as follows:

- calculation of a frame energy,
- Hamming window filtering,
- FFT calculation,
- frequency conversion to the mel scale,
- frequency filtering by 26 triangular filters (spread over the whole frequency range up to the Nyquist frequency),
- calculation of 12 MFCC parameters,
- calculation of the first and second derivatives.

2.3. MODELS TRAINING

In the first stage the prototype model was created. In the work, the decision was made to use the most common three-state left-to-right HMM. The prototype model was the same for all phonemes.

Two approaches are used to initialize the prototype model in HTK. One of them is called the “flat start”. In that method the HCompV tool is used. It calculates global mean and variance of all training data and then inserts these values to all states of all models.

Another method is an individual initialization of every state using the HInit tool. It tries to adjust each state parameters in such a way that it best matches the input data. HInit works as follows. In the first stage it uniformly segments the training data and associates successive segments with successive states. If the state consists of one mixture component, simply the mean and variance values are calculated from the data vectors of the segment data associated with the state. If the state consists of more than one mixture component then at first the K-means algorithm is used to cluster training data vectors and, after that, means and variances are calculated for each mixture component. The values of the transition matrix remain unchanged at this stage.

The second stage only differs from the previous one in that the Viterbi algorithm is used instead of uniform segmentation. The stage is repeated until the likelihood change of the model for training samples in two consecutives steps falls below the earlier defined threshold or if the step number of the iteration achieves the limit.

The HInit method gives usually better results but it requires properly segmented and labelled training data. Especially, in the case of subword models such as phoneme models, it is inconvenient to prepare a reasonable training set for initialization. For that reason the HCompV method was used in the work:

HCompV -C cfg -f 0.01 -m -S train.scp hmm

where: ‘cfg’ defines the parameterization of audio files, ‘f 0.01’ sets the variance floor value, ‘m’ forces updating model means, train.scp is the list of audio files, ‘hmm’ is the model to initialize.

The next step is the model training. There are two tools for training contained in HTK: the HRest and HERest. HRest works similar to HInit except that it uses the Baum-Welch (forward-backward) algorithm. HRest, in contrast to HInit, finds the probability of being of a model in each state at each frame time and then updates model parameters. HInit uses only information about the most probable state sequence of the model. It is assumed that the Baum-Welch algorithm is “better” in the case of phone-based HMMs since there are no strict boundaries between phones in real speech [6]. HRest is designed to reestimate parameters of separated models so it requires separate training data sets.

The second tool, HERest works in an another way. It simultaneously updates parameters of all models using all of the training data. It utilizes the concept of the “embedded training”. At the start it loads the complete set of HMMs into memory. Next, for each training file, HERest constructs a composite HMM consisting of all models corresponding to the transcription of the input utterance. All of the training files are then processed in one turn using the Baum-Welch algorithm. After that new parameters are estimated from the weighted sums and a new, updated set of HMMs is obtained.

In the work two different silence models were used: the ‘sil’ model (silence model) and the ‘sp’ model (short pause model). The ‘sil’ model looked like other phoneme model and was created to handle longer periods of silence. Such silence occurs at sentence borders or inside a sentence, at punctuation marks for example.

In a speech, the period of a silence between consecutive words is often very short or even non-existent. To handle that the ‘sp’ model was used. It had three states and only one was emitting. The emitting state was tied (set equal) to the centre state of the ‘sil’ model. There was also the transition added between the start and end states (non-emitting states). The model with such transition is called a tee-model and enables modelling optional transient effects (short pauses).

The training process was divided into two phases. In the first phase all models except the “sp” one were trained. The models were trained using HERest command in the following form:

HERest -C cfg -I mlf -S train.scp -H input_hmm -M output_hmm phonelist

where: ‘cfg’ defines the parameterization of audio files, ‘mlf’ is the MLF file with a phonetic transcription of training utterances, ‘train.scp’ is the list of training files, ‘input_hmm’ is the list of HMMs definitions, ‘output_hmm’ is the output file for the updated HMMs, ‘phonelist’ is the list of models to train.

After the second step of iteration the ‘sil’ model was modified. There were additional transitions added: from state 2 to state 4 and from state 4 to state 2. These transitions made the ‘sil’ model more robust and allowed it to absorb the background noise without transition to the next one (to the following phoneme). At this stage the ‘sp’ model was also created. The center state and the transition matrix of the ‘sil’ model were manually copied to the ‘sp’ model. The transition matrix was properly modified according to the earlier ‘sp’ model description. Next center states of the ‘sp’ and ‘sil’ models were tied.

After modification of silence models, HERest was run again. The number of iteration steps was seven. The used command was:

HERest -C cfg -I mlf -S train.scp -H input_hmm -M output_hmm phonelist1

where: ‘phonelist1’ is the file with the list of models to train with the additional ‘sp’ model included.

2.4. RECOGNITION TESTS

The recognition database consisted of 39 context-independent models: one model per one phoneme and two additional silence models. The recognition tool was HVite.

HVite requires some additional information about language rules i.e. a network model. Because of the fact that in the work only phoneme transcription was required the network model should define only allowed phoneme sequences. In the work the simplest network model was used - a phoneme loop, where every phoneme may follow another one without a limit. The network definition file looked like:

$phoneme=a|a~|b|c|cz|ci|d|dz|dzi|e|drz|e~|f|g|h|i|j|k|l|l~|m|n|ni|o|p|r|s|sz|si|t|u|w|y|z|zi|rz|sil;
(sil <$phoneme> sil)

HVite accepts only Standard Lattice Format (SLF) so the above network definition file was transformed using HParse to a lattice file:

HParse network lattice

After that recognition was accomplished. In order to obtain a phonetic transcription the following command was used:

HVite -C cfg -H hmmdefs -S inputlist -i results -w lattice -p -5.0 vocabulary phonelist1

where: 'cfg' defines the parameterization of audio signal, 'hmmdefs' contains the trained hmm models, 'inputlist' is the list of tested audio files, 'results' is the file with the obtained phonetic transcription of the recognized utterances, 'lattice' is the phoneme network, 'p -5.0' is the word insertion penalty, 'vocabulary' is the pronunciation-phoneme map, 'phonelist' is the list of models used during recognition.

In the list of HVite parameters there is a "vocabulary" file. The same word can have even several ways of pronunciation. In order to the interpret recognized sequence properly HVite needs to have a vocabulary file that describes pronunciations of all recognized words.

In the work the vocabulary file was used only to replace the sequence of model names into the sequence of phoneme symbols.

For evaluation purposes two common percentage parameters were used:

Correctness: $$C = \frac{H}{N} * 100\%$$

Accuracy: $$A = \frac{H - I}{N} * 100\%$$

where: H - number of properly detected phonemes, N - total number of phonemes, I - the number of insertion errors (phonemes mistakenly detected).

At first, only phoneme recognition level was checked. HVite has several parameters which can influence the recognition quality. One of them is the "word insertion penalty" option (the '-p' option) which affects the number of deletions (not detected phonemes) and insertion errors. In order to improve the recognition quality several values of that option were tested. As a result, the value of p=-5 was selected. It balanced deletion and insertion errors. Finally, the achieved recognition quality was: correctness 51%, accuracy 33% (properly recognized phonemes were 1157 of the total number 2272, 406 deletions, 403 insertions and 709 substitutions).

2.5. REPETITION RECOGNITION

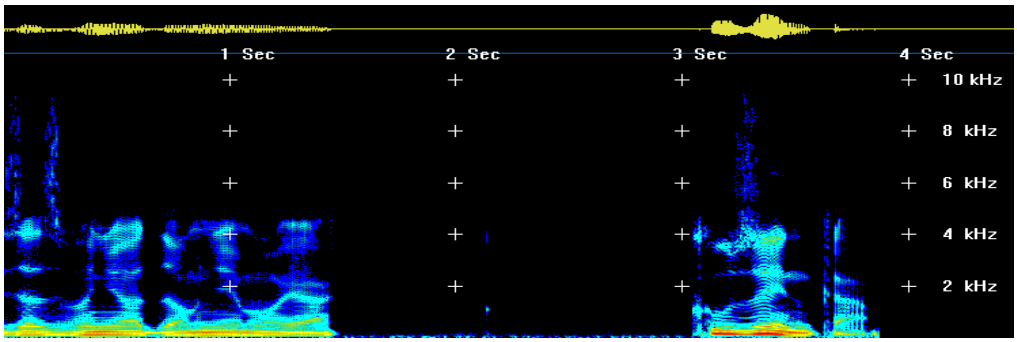
The detection idea of phoneme repetitions was to generate a phoneme transcription from the inspected utterances and then check whether there are some sequences of phonemes compounded from plosives (stop consonants) and silence. Such detected sequences could be treated as repetitions. Additionally, dysfluent phonemes can be identified.

The tested set consisted of 79 utterances and 20 of them included plosives phoneme repetitions (k,d,t,p). The rest was fluent.

Fig. 1 shows the example of recognition of the utterance "za'dz'woniładomniemoja_k_kuzynka". In the spectrogram one can see that the dysfluency begins at 01:30 s and ends at 03:00 s and lasts 01:30 s. It is mainly the silence but in the middle (at 02:07s) some articulation is visible - this is the phoneme 'k'. The part of the recognized phoneme sequence was:

…
1428 ms 148 ms _
1486 ms 2101ms _
2101 ms 2147 ms k
2147 ms 3006 ms _
3006 ms 3123 ms k
…

One can see that the obtained transcription ideally fits the spectrogram, the non-fluency can be properly detected and the disturbed phoneme can be also identified.



a)
b) za'dz'woniładomniemoja_k_kuzynka
c) _zazl~rmjadomjml~lja__k_kuzye~ka_

Fig. 1. The example of recognition of the Polish utterance “za'dz'woniładomniemoja_k_kuzynka” a) spectrogram [4], c) orginal transcription, b) obtained transcription.

The test results are presented in two tables. Table 2 presents the level of phoneme repetitions recognition without taking the identification of a phoneme into account. The obtained values of correctness and accuracy are very good.

Table 3 presents the level of phoneme repetitions recognition taking the identification of a phoneme into account. In this case the lack of proper identification of the repeated phoneme is treated as an error. Correctness and accuracy values are rather poor. This could be caused by quality of training material because it was recorded in a real environment from a stuttering man without obeying specials conditions.

Table 2. The level of repetitions recognition.

	Total number	Correctly detected	Not detected	Incorrectly detected	Correctness	Accuracy
Repetitons	18	16	1	1	89%	83%

Table 3. The level of repeated phonemes identification.

	Total number	Correctly detected	Incorrectly detected	Correctness	Accuracy
Repeated phonemes	16	8	8	50%	0%

3. SUMMARY

Automatic detection of speech disorders can have very practical application. In the work the HTK toolkit was used to build the simple monophone recognition system. In conjunction with the Perl scripts, it gave possibility to recognize speech disorders like repetitions. Despite its simplicity, the performed tests gave promising results. Such automatic systems can support speech therapists so the therapy process of stuttering people can be easier, faster and more objective.

BIBLIOGRAPHY

[1] ACHIBALD L., DE NIL L.F., The relationship between stuttering severity and kinesthetic acuity for jaw movements in adults who stutter, Journal of Fluency Disorders, Vol. 24(1), Elsvier, 1999, pp. 25-42.

[2] CZYŻEWSKI A., KACZMAREK A., KOSTEK B., Intelligent processing of stuttered speech, Journal of Intelligent Information Systems, Vol. 21(2), Kluwer Academic Publishers, The Nederlands, 2003, pp. 143-171.

[3] GAJECKI L., TADEUSIEWICZ R., Modeling of Polish Language for Large Vocabulary Computer Speech Recognition, Speech and Language Technology, Vol. 11, Poznań, 2008, pp. 65-70.

[4] HORNE R. S., Spectrogram for Windows, ver. 3.2.1.

[5] HOWELL P., SACKIN S., Automatic recognition of repetitions and prolongations in stuttered speech, Proceedings of the First World Congress on Fluency Disorders, 1995, pp. 372-374.

[6] http://htk.eng.cam.ac.uk/docs/docs.shtml.

[7] JASSEM W, Podstawy fontetyki akustycznej, PWN, Warszawa, 1973, (in Polish).

[8] SMOŁKA E., KUNISZYK-JÓŹKOWIAK W., SUSZYŃSKI W., DZIEŃKOWSKI M., SZCZUROWSKA I., Speech Nonfluecy Recognition in Two Stages of Kohonen Networks, Structures-Waves-Human Health, Zakopane, 2004, pp. 139-142.

[9] SZCZUROWSKA I., KUNISZYK-JÓŹKOWIKAK W., SMOŁKA E., The Application of Kohonen and Multilayer Perceptron Networks in the Speech Nonfluecy Analysis, Archives of Acoustics, Vol. 31, 2006, pp. 205.

[10] TADEUSIEWICZ R., Speech Recognition with Application of Neural Networks, Seminar of Polish Phonetical Society, Warszawa, 1994, pp. 137-150.

[11] WIŚNIEWSKI M., KUNISZYK-JÓŹKOWIAK W., SMOŁKA E., SUSZYŃSKI W., Automatic detection of disorders in a continuous speech with the Hidden Markov Models approach, Advances in Soft Computing 45, Computer Recognition Systems 2, Vol. 45, Springer-Verlag, Berlin Heildelberg, 2008, pp. 445-453.

[12] WIŚNIEWSKI M., KUNISZYK-JÓŹKOWIAK W., SMOŁKA E., SUSZYŃSKI W., Improved approach to automatic detection of speech disorders based on the Hidden Markov Models approach, Journal of Medical Informatics and Technologies, Vol.15, Computer Systems Dep., University od Silesia, Poland, 2010, pp. 145-152.

[13] WSZOŁEK W., TADEUSIEWICZ R. The Evaluation of Effectiveness of Various Neural Network Types in Pathological Speech Analysis, XLVII Open Seminar on Acoustics OSA`2000, Vol. 2, Rzeszów – Jawor, 2000, pp. 721-728.