EURECOM
Sophia Antipolis

Institut Eurécom
Department of Corporate Communications
2229, route des Crètes
B.P. 193
06904 Sophia-Antipolis
FRANCE

Research Report RR-06-182

DASR: A Diagnostic Tool For Automatic Speech Recognition

December 6, 2006

Milos Cernak

Tel : (+33) 4 93 00 82 37
Fax : (+33) 4 93 00 82 00
Email : Milos.Cernak@eurecom.fr

[1] Institut Eurécom's research is partially supported by its industrial members: Bouygues Téécom, France Téécom, Hitachi Europe, SFR, Sharp, ST Microelectronics, Swisscom, Texas Instruments, Thales.

Abstract

In this report we present a Diagnostic tool for ASR systems (DASR). The aim was to develop a tool capable to perform statistical analysis of output of ASR decoding process. Many error patterns in the output might be observable directly by humans, but if number of tracking variables (possible causes of errors) is very high, the task for humans becomes too complex. Machines are able to process as much variables as necessary, and performs a statistical analysis on data as well. We discuss design and implementation of the tool.

In addition, we present an example of usage of the tool. This is an explorative study of diagnostics of speech recognition for finding subsets of features that are most informative in terms of incorrect speech recognition, if variable speech is recognized. The impact on both MFCC and PLP features is investigated.

1 Introduction

Most of research in the field report results in terms of ever-lower WER acquired over some baseline, leaving questions about the causes of failures open. Evaluation of recognizer performance is usually expressed in terms of few figures like WER and confusion matrix. Diagnostics complements the evaluation. While evaluation is defined as an assessment of the system, measuring some parameters of the system, diagnostics is a computing mechanism to identify faults of the system. In other words, diagnostics is the identification and more challenging, the understanding of incorrect speech recognition. Diagnostics of speech recognition should provide error patterns of the decoding process as well as of the training process.

Recognition may be studied in detail considering different linguistic or phonetic properties [1]. The recognition results are usually identified using the acoustic-phonetic classes [2, 3]. Some authors go further and try to find a reason of phoneme confusion, or even their deletions and insertions. In a recent work [4], authors explored some articulatory properties of confused consonants. Comparing human and computer speech recognition, they concluded, that voicing information should actually be used for better performance of machine speech recognition.

In our work we use a decision tree analysis, following work of [5, 6, 7]. The idea is to incorporate statistics of building decision trees for finding factors that cause the systematic recognition errors. We are motivated by development of Lin Chase's CMU Error Region Analysis (ERA) tool[1], which was our starting point for further consideration about the task of ASR diagnosis. The aim of our work was also to develop a tool capable to perform statistical analysis of output of ASR decoding process. The tool was designed to use within European DIVINES project[2], using TORCH machine-learning library [8] and OLLO speech database [9], but it could be easily adapted for other system and task setups.

The report is structured as follows. Section 2 introduces DASR tool. Next section 3 shows an example of its usage focusing on an analysis of standard feature sets (MFCC and PLP) of ASR systems. Section 4 describes comparison of the tool with other tools and section 5 concludes the report.

2 Main Concepts of DASR Tool

DASR tool is an implementation of decision tree analysis in the context of ASR diagnosis task. The process of diagnosis is shown in Fig. 1. It is necessary to provide to DASR the following data:

- ASR output in the form of reference (henceforth REF) and hypothesized (henceforth HYP) sequences.

- Feature representations as possible causes of errors.

[1] http://www.cs.cmu.edu/afs/cs/user/lindaq/ERA/

[2] http://www.divines-project.org/

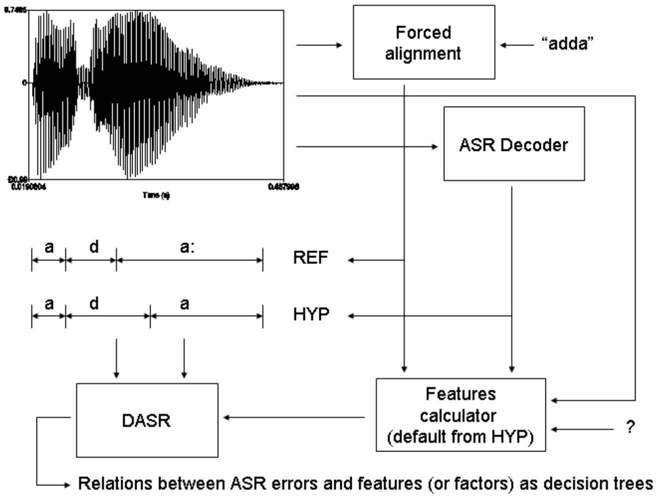

0.7485
0
0.0190604
0.487998
Time (s)
Forced alignment
"adda"
ASR Decoder
a
d
a:
REF
a
d
a
HYP
DASR
Features calculator (default from HYP)
?
Relations between ASR errors and features (or factors) as decision trees

Figure 1: Overview of diagnostic process. The question mark represents any other inputs in addition to acoustic representation for feature calculation, such as phonological and/or articulatory information.

It is important to note here that DASR tool is able to perform statistical analysis in order to look for relation between ASR errors and possible causes of errors (factors represented by speech features), but the specification of factors must be done by user. In other words, the user has to have some intuition about possible causes of errors. These speech features[3] might be seen in three levels: the discrete phonological representation of an utterance, the acoustic pattern that results from the utterance, and the articulatory gestures that create the links between the phonological and acoustic representations.

We designed and implemented the DASR (Diagnostics of ASR) tool in MATLAB environment. We did it purposely, because this environment supports many publicly available algorithms for easy speech feature extraction, that is necessary for features calculator block (see Fig. 1). Our speech recognition decoder (see Section 3.1 for more details) generates either ERA_IN files or CTM files, witch both store the reference and decoded phoneme sequences with time boundaries (see a description of file formats in appendixes A and B respectively). This gives users of the DASR Tool an opportunity to use also Lin Chase's CMU Error Region Analysis (ERA) tool, and scoring the output of speech recognizers via the NIST `sclite()` program. We found interesting to use both programs during our work on speech recognition diagnostics.

[3] We use the term 'features' in diagnostic process for various representations of a part of speech vaweform; not be confused with the features extracted from the waveform decoding purposes.

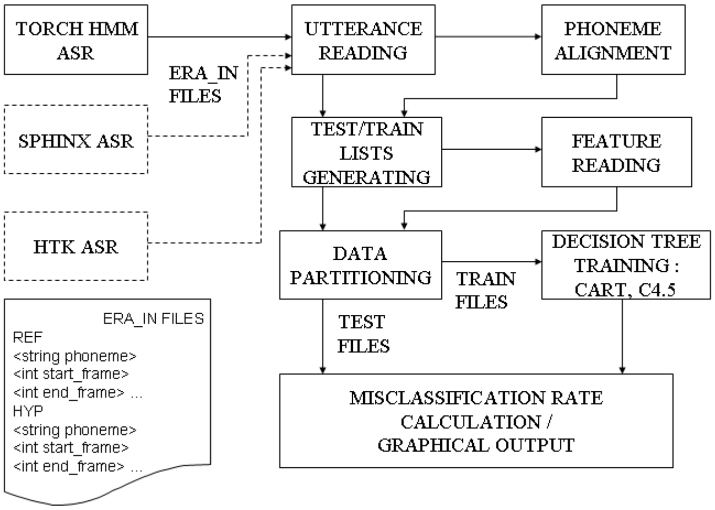


Figure 2: Data flow in DASR. The schema highlights the main functional modules. The input is provided by speech decoder, and the tool further processes the data using decision tree analysis. Dashed boxes are optional, showing that DASR tool should be independent of used ASR system.

The overview of our tool is depicted in Fig. 2. Data processing can be split into following tasks:

1. Load ASR data. The output files of the decoder are converted and stored in an internal format, which stores all HYP and REF sequences.

2. Alignment of the sequences. The initial list is split in two parts, the first containing REF sequences and the second part of HYP sequences, which are aligned using maximal substring matching [10].

3. Merge ASR and aligned data. Here a data list is generated, which contains all available data. Data structure is shown in Fig. 3. User can choose a predictee (predicted variable) for decision tree analysis.

4. Generation of training and testing lists for decision tree analysis. The training and testing files for decision tree analysis are generated.

5. Load parametrization of HYP sequence. The features (at the acoustic, phonetic, phonologic level) are loaded or calculated. Any feature calculation has to be done individually; it is not included in DASR tool.

6. Training of decision trees. Here decision tree analysis is performed. The primary technique for the analysis that the tool supports, is the CART technique described in [11]. In addition, C4.5 technique [12] is supported, as its importance for diagnostic purposes has been already shown in [13].

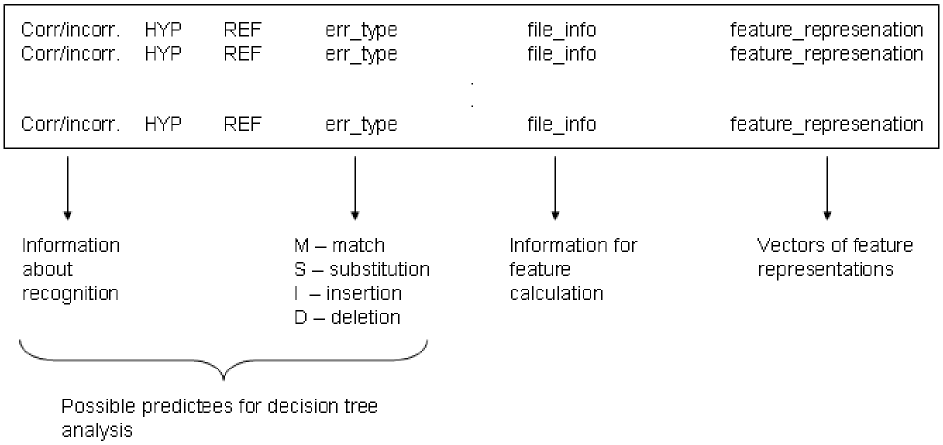

Corr/incorr. HYP REF err_type file_info feature_represenation
Corr/incorr. HYP REF err_type file_info feature_represenation
Corr/incorr. HYP REF err_type file_info feature_represenation
Information about recognition
M – match
S – substitution
I – insertion
D – deletion
Information for feature calculation
Vectors of feature representations
Possible predictees for decision tree analysis

Figure 3: Structure of a data list produced by DASR tool. First part contains predictees for decision tree analysis, second part is consisted of information about features.

7. Testing and printing of decision trees. Misclassification scores may be calculated, and trained trees may be printed in a text tabular fashion.

To help categorize the errors, we use similar concepts as in [6] and [14]. Let t_{w_i} denote the start frame of the i-th phone in a transcription, then the central position of the i-th phone can be written as:

$$c(w_i) = (t_{w_{i+1}} - t_{w_i})/2 \tag{1}$$

Using maximum substring matching algorithm we assign to each HYP phoneme one of the following categories: match, substitution, insertion or deletion. Using this information we add a label about correct or incorrect decoding as well. In addition, using two aligned sequences $\widehat{\mathbf{w}}$ for decoded sequence and $\mathbf{w}$ for reference sequence, we define w_j as the REF phoneme to the HYP phoneme $\widehat{w}_i$ in the following way:

1. If $\widehat{w}_i$ has a label match or substitution, we define w_j as its REF phoneme if $j = i$. Here j is an index to the REF sequence and i is an index to the HYP sequence.

2. If $\widehat{w}_i$ has a label insertion or deletion, we define w_j as its REF phoneme if:

$$t_{w_j} < c(\widehat{w}_i) <= t_{w_{j+1}}, \tag{2}$$

where j is an index to the REF sequence, i is an index to the HYP sequence, t_{w_j} is the start frame of the REF phoneme w_j, and $c(\widehat{w}_i)$ is the central position of the HYP phoneme $\widehat{w}_i$.

3 How To Use DASR Tool: An Example

Recently we have introduced a concept of Phoneme Diagnostic Trees (PDTs) [15]. For each basic phoneme used in an ASR system a PDT is constructed, which links incorrect recognitions of the phoneme with a-priori specified sources of errors (factors). Decision trees PDTs then describe how a given input (reference phoneme) can correspond to specific outputs (decoded phoneme), as a function of these factors. These PDTs can be generated by DASR tool, if the user chooses from generated data list (see Fig. 3) only incorrect recognition items and the REF labels as predictees for the analysis.

In addition, we present here an another example. There is an extensive literature on acoustic features for ASR and their selection (see e.g. [16, 17]), which is still difficult task. The aim of this example is to get better understanding of the performance of the different feature sets and their subsets in the terms of speech variabilities.

In speech recognition, speech variability is one of the major error sources. Speech variabilities may be classified to the two main categories: extrinsic variabilities are due to the environment (noise, telecommunication channels), and intrinsic variabilities that convey information about the speaker himself (gender, age, social and regional origin, health and emotional state) [1]. There is also a well studied impact of stressed speech on speech and speaker recognition [18]. Stress in this context refers to speech produced under cognitive, physical, emotional stress, and stress due to presence of noise (known as the Lombard effect). Research on impact of intrinsic speech variabilities and stressed speech on speech recognition is overlapped. We have recently found a link between intrinsic speech variations and emotional speech (as a kind of stressed speech) [19].

Within the European DIVINES project (divines-project.org) we study speech recognition deficiencies in dealing with speech recognition variabilities. The ultimate goal would be to achieve better understanding of source of errors, or a signal modeling framework and robust features which are immune to the intrinsic speech variations. In the following sections, we are focused on an analysis of standard feature sets (MFCC and PLP) of ASR systems, exploring impact of intrinsic speech variabilities on speech recognition.

3.1 Used Database and ASR System

We use the OLLO database, which has been recorded for the purpose of study of speech recognition deficiencies in dealing with speech intrinsic variabilities. The database is designed for recognition of individual phonemes that are embedded in logatomes, specifically, CVC and VCV sequences. Several intrinsic variabilities in speech are represented in OLLO, by recording from 40 speakers from four German dialect regions, and by covering three speaker-dependent variabilities: gender, age and dialect, and six speaker-independent variabilities: fast, slow, lound, quit, question and statement speaking styles. We used NO-accent training and testing parts of OLLO database.

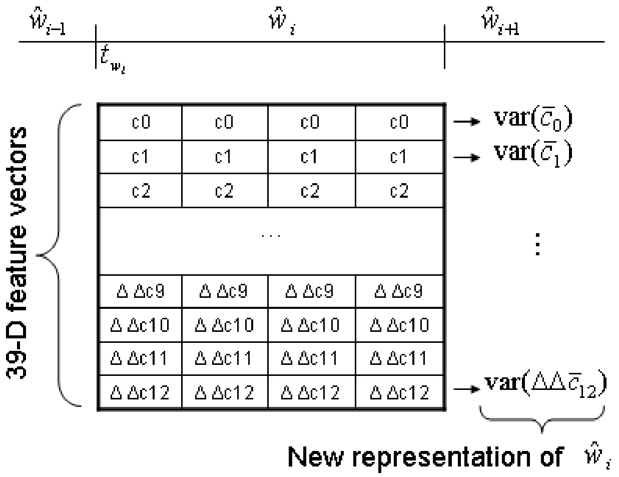
Hidden Markov Models (HMM) and Gaussian Mixture Models (GMM) based speech recognition system is trained using public domain machine-learning library TORCH on the training set that consists of 13446 logatome utterances. Three states left-right HMM models were trained for each of the 26 phonemes in the OLLO database including silence as well. Gaussian mixture models with 17 Gaussians per state and diagonal covariance matrices were used to model the emission probability densities of the 39 dimensional feature vectors - 13 cepstral coefficients and their derivatives (Δs) and double derivatives ($\Delta\Delta$s). The phoneme HMMs are connected with no skip. We extended the TORCH library in a package of calculation and storage of feature data, necessary for further statistic processing. The decoder collects the feature data by running on the testing set that consists of 13466 logatome utterances. We trained and tested two ASR systems, one with MFCC feature set and the second with the PLP front-end. All the features were calculated using HTK hcopy tool. We calculated MFCC vectors every 10 msec using windows of size 25 msec. The same settings were applied also for calculation of PLP vectors, we only used power rather than the magnitude of the Fourier transform in the binning process. Average phoneme recognition performance of the ASR systems on this task was 76.06 % (the lowest accuracy had recognition of fast speech: 71.94 %, and the highest accuracy had speech with statement style: 80.48 %). The MFCC features performed slightly better than PLP features (all our experiments were done on clean speech).

During the decoding process, both correct and incorrect decodings (cases in the terminology of decision tree analysis) are collected. The REF sequence is acquired by Viterbi forced alignment. At the end of the Viterbi computation for the last frame of the utterance the aligner stores the phone assignments to the frames, along with the actual scores associated with each segmentation.

3.2 Decision Tree Analysis

Decision tree analysis is performed based on the observation vectors of the MFCC and PLP coefficients (c_0, c_{1-12}, their derivatives and double derivatives). Motivated by [20], we calculated variance of the feature vectors for each HYP phoneme. Fig. 4 overviews the calculation of the 39-D phoneme feature representation used for the further analysis.

Variance of speech features is calculated for each of HYP phonemes. This new 39-D parametrization is stored in the list (one item for one HYP phoneme), together with the labels about correct or incorrect decoding. These labels are later predictees for decision tree training process (see Fig. 3, first column). We used CART technique to create six decision trees, one for each speech variability (5 variabilities plus 1 normal, statement style, speech). All the presented results in this paper were got using stopping grow criterions of minimal 10 of the cases in a terminal node and minimal entropy gain of 3%. Splitting of the correct/incorrect cases during the training was done using questions about variances of features.



c0	c0	c0	c0
c1	c1	c1	c1
c2	c2	c2	c2
...			
ΔΔc9	ΔΔc9	ΔΔc9	ΔΔc9
ΔΔc10	ΔΔc10	ΔΔc10	ΔΔc10
ΔΔc11	ΔΔc11	ΔΔc11	ΔΔc11
ΔΔc12	ΔΔc12	ΔΔc12	ΔΔc12

$\hat{w}_{i-1}$
$\hat{w}_i$
$\hat{w}_{i+1}$
t_{w_i}
39-D feature vectors
$\rightarrow \mathrm{var}(\bar{c}_0)$
$\rightarrow \mathrm{var}(\bar{c}_1)$
⋮
$\rightarrow \mathrm{var}(\Delta\Delta\bar{c}_{12})$
New representation of $\hat{w}_i$

Figure 4: Calculation of 39-D phone feature representation used in decision tree analysis. The picture shows an example of the calculation for four frames of the HYP phoneme $\widehat{w}_i$.

Variability	Misclass. rate	Features
Fast	16.39 %	c_{12}
Slow	24.53 %	$c_{12}, c_0, \Delta\Delta c_8, c_5, \Delta c_3$
Loud	18.26 %	c_{12}
Quiet	27.80 %	c_{12}
Question	27.43 %	$c_{12}, c_8, c_9, \Delta c_9, \Delta\Delta c_{10}$
Normal	15.27 %	c_{12}

Table 1: Major MFCC coefficients selected

3.3 Results

In this section, we present major results obtained from this study. We investigated both MFCC and PLP features. We went over the trained trees, following paths leading to the most probable classification of incorrect decodings. We collected all the features associated with these paths. We can interpret these features as most significant features for prediction of incorrect decoding. The results for MFCC and PLP front-ends are shown in the tables 1 and 2, respectively. Decision trees for normal speech (trained on both MFCC and PLP features) have the lowest misclassification rates (the lowest estimated accuracies of trained classifiers). This implies that building classifiers/predictors for correct/incorrect recognition for variable speech is more difficult. In addition, PLP decision trees have higher misclassification rates than MFCC trees. We observed that it follows the trend of lower ASR performance if PLP features are used (in clean speech).

Variability	Misclass. rate	Features
Fast	18.44 %	c_{12}
Slow	26.49 %	$c_{12}, c_0, \Delta\Delta(c_{12}, c_0), \Delta c_{12}$
Loud	22.37 %	$c_{12}, c_0, c_7, \Delta\Delta c_4, \Delta c_{11}$
Quiet	31.82 %	c_{12}, c_0
Question	26.37 %	$c_{12}, c_0, \Delta\Delta c_{12}, \Delta c_6, c_6$
Normal	17.85 %	c_{12}

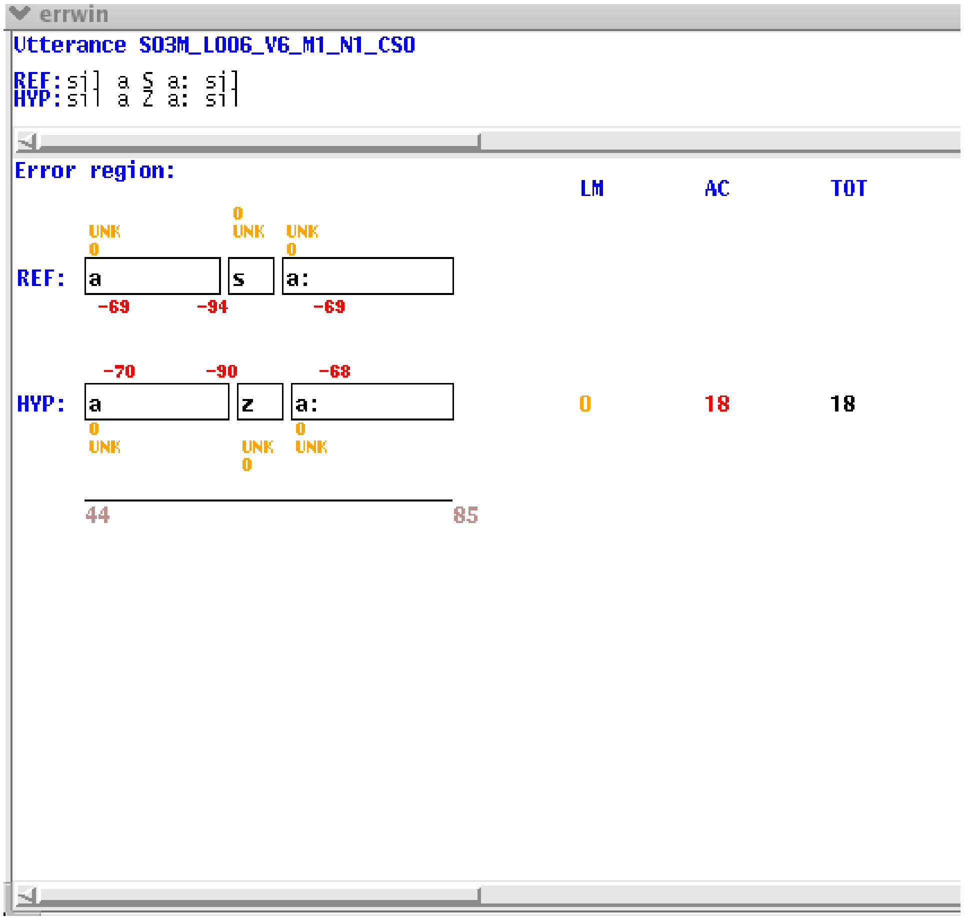
Table 2: Major PLP coefficients selected

In [21, 22] the authors shows that the lower quefrency coefficients generally have higher F-ratio (a measure of separability between multiple speech classes) and should therefore offer better class separation and so better ASR performance. Arslan and Hansen [23] have also confirmed, that coefficients in the middle of quefrency region are the most relevant for dialect classification. Our findings imply that upper quefrency region (plus deltas and double deltas) is the most informative for predicting incorrect speech recognition. The most informative coefficient across all the variable speech recognition for this prediction was in our study c_{12} coefficient. For slow and questioning styled speech also dynamic features were found most informative. Dynamic features were found relevant also for loud speech in using PLP frond-end.

The general conclusion of this study is that the upper quefrency region and less middle region are the most informative for predicting incorrect speech recognition. Discarding the higher cepstral coefficients is sometimes normal practice in ASR. We confirmed that these coeffiences are problematic. In addition, we proposed the diagnostic technique for exact specification of problematic coefficients. Some previous works confirmed different contribution of quefrency regions to recognition of stressed speech [23, 24]. New frequency scales have been there proposed, which are less sensitive to variations caused by stress without degrading the performance of neutral speech recognition. Having results of our study we confirm that upper quefrency region is also very important in terms of incorrect speech recognition.

4 Comparison With Other Tools

To our knowledge there is no other tool designed specifically for ASR diagnosis. However, it is worth to study Lin Chase's PhD thesis and her Error Region Analysis ERA tool [6]. Fig. 5 shows graphical presentation of some error regions as specified automatically by ERA tool. The analysis clearly separates contributions of acoustic and language modeling. Similarly as Eide [5], Chase used decision tree analysis for further processing. As features she used representation often described in works that deal with confidence measures for ASR (see e.g. [25, 26, 27]).



```
Utterance S03M_L006_V6_M1_N1_CSO

REF:sil a S a: sil
HYP:sil a Z a: sil

Error region:
```

Figure 5: An example of Lin Chase's graphical presentation of errors done by ASR. HYP acoustics here is better than REF acoustics. According to [6] the reason might be that (a) speech is not modeled well (this includes e.g. fast speech) and (b) there is the presence of confusions between acoustic models that allow data that actually represent one phone to be decoded as another with a high score.

5 Conclusion

We have presented DASR tool for making diagnostics of automatic speech recognition systems. The aim was to contribute to ASR diagnostics, as an important issue toward better understanding of causes of ASR errors. We tried to make the tool platform independent, and independent of used ASR system as well. The tool was designed as an evolution of current published approached to ASR diagnostics, with emphasis to be redistributed, used, and last but not least contributed by other researchers in the field.

6 Acknowledgments

I would like to express my best thanks to Prof. Christian Wellekens who supervised this work, and who helped me a lot on this project.

This work has been supported by the EU 6^{th} Framework project DIVINES under the contract number IST-2002-002034. The views expressed here are those of the authors only. The Community is not liable for any use that may be made of the information contained therein.

References

[1] M. Benzeghiba, R. De Mori, O. Deroo, S. Dupont, D. Jouvet, L. Fissore, P. Laface, A. Mertins, C. Ris, R. Rose, V. Tyagi, and C. Wellekens, "Impact of variabilities on speech recognition," in *SPECOM'2006*, Saint-Petersburg, Russia, June 2006, pp. 3–16.

[2] G. Chollet, A. Astier, and M. Rossi, "Evaluating the performance of speech recognisers at the acoustic-phonetic level," *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP '81*, vol. 6, pp. 758–761, 1981.

[3] M. Hunt, "Speech recognition, syllabification and statistical phonetics," in *Proc. of ICSLP*, Jeju Island, Korea, October 2004.

[4] Bernd Meyer, Thorsten Wesker, Thomas Brand, Alfred Mertins, and Birger Kollmeier, "A human-machine comparison in speech recognition based on a logatome corpus," in *ITRW on Speech Recognition and Intrinsic Variation*, May 2006.

[5] E. Eide, H. Gish, P. Jeanrenaud, and A. Mielke, "Understanding and improving speech recognition performance through the use of diagnostic tools," in *International Conference on Acoustics, Speech, and Signal Processing, ICASSP-95*, May 1995, vol. 1, pp. 221–224.

[6] Lin Chase, *Error-Responsive Feedback Mechanisms for Speech Recognizers*, Ph.D. thesis, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, 15213, USA, April 1997.

[7] G. Greenberg and S. Chang, “Linguistic dissection of switchboard-corpus automatic speech recognition systems,” in *Proc. of ITRW on Automatic Speech Recognition: Challenges for the new Millenium*, Paris, France, 2000, pp. 195–202.

[8] R. Collobert, S. Bengio, and J. Marithoz, “Torch: a modular machine learning software library,” Tech. Rep. IDIAP-RR 02-46, IDIAP, 2002.

[9] T. Wesker, M. Meyer, K. Wagener, J. Anemuller, A. Mertins, and B. Kollmeier, “Oldenburg logatome speech corpus (ollo) for speech recognition experiments with humans and machines,” in *Interspeech 2005*, September 2005, pp. 1273–1276.

[10] X. Huang, A. Acero, and H-W Hon, *Spoken Language Processing*, Prentice Hall PTR, New Jersey, 2001.

[11] L. Breiman, J. Friedman, Ch J. Stone, and R. A. Olshen, *Classification and Regression Trees*, Chapman & Hall/CRC, New York, 1983.

[12] J. R. Quinlan, *C4.5: Programs for Machine Learning*, Morgan Kaufmann Publishers, San Meteo, CA, USA, 1993.

[13] W. Buntine, “Tree classification software,” in *Technology 2002: The Third National Technology Conference and Exposition*, Baltimore, USA, December 1992.

[14] L. Zhang and S. Renals, “Phone recognition analysis for trajectory hmm,” in *Interspeech2006 ICSLP*, September 2006.

[15] M. Cernak and C. Wellekens, “Diagnostics of speech recognition using classification phoneme diagnostic trees,” in *CI 2006 (Special Session on NLP)*, San Francisco, CA, USA, November 2006.

[16] J. W. Picone, “Signal modeling techniques in speech recognition,” *Proceedings of the IEEE*, vol. 81, no. 9, pp. 1215–1247, 1993.

[17] E. L. Bocchieri and J. G. Wilpon, “Discriminative feature selection for speech recognition,” *Computer Speech and Language*, vol. 7, no. 3, pp. 229–246, July 1993.

[18] John H. Hansen, “Speech under stress,” in *Interspeech2006 ICSLP*, Pittsburgh PA, USA, September 2006.

[19] M. Cernak and C. Wellekens, "Emotional aspects of intrinsic speech variabilities in automatic speech recognition," in *SPECOM'2006*, Saint-Petersburg, Russia, June 2006, pp. 405–408.

[20] D. X. Sun and L. Deng, "Analysis of acoustic-phonetic variations in fluent speech using timit," in *International Conference on Acoustics, Speech, and Signal Processing, ICASSP-95.*, Detroit, USA, 1995, vol. 1, pp. 201–204 vol.1.

[21] K. K. Paliwal, "Dimensionality reduction of the enhanced feature set for the hmm-based speech recognizer," *Digital Signal Processing*, vol. 2, no. 3, pp. 157–173, July 1992.

[22] S. Nicholson, B. Milner, and S. Cox, "Evaluating feature set performance using f-ratio and j-measures," in *EUROSPEECH 97*, Rhodes, Greece, September 1997, pp. 413–416.

[23] Levent M. Arslan and John H. L. Hansen, "A study of temporal features and frequency characteristics in american english foreign accent," *The Journal of the Acoustical Society of America*, vol. 102, no. 1, pp. 28–40, 1997.

[24] S. E. Bou-Ghazale and J. H. L. Hansen, "A comparative study of traditional and newly proposed features for recognition of speech under stress," *Speech and Audio Processing, IEEE Transactions on*, vol. 8, no. 4, pp. 429–442, 2000.

[25] T. Schaaf and T. Kemp, "Confidence measures for spontaneous speech recognition," in *Acoustics, Speech, and Signal Processing, 1997. ICASSP-97., 1997 IEEE International Conference on*, 1997, vol. 2, pp. 875–878 vol.2.

[26] S. O. Kamppari and T. J. Hazen, "Word and phone level acoustic confidence scoring," in *Acoustics, Speech, and Signal Processing, 2000. ICASSP '00. Proceedings. 2000 IEEE International Conference on*, 2000, vol. 3, pp. 1799–1802 vol.3.

[27] D. A. G. Williams, *Knowing What You Dont Know: Roles for Confidence Measures in Automatic Speech Recognition*, Ph.D. thesis, Department of Computer Science, University of Sheffield, May 1999.

A ERA_IN File Format

Reproduction of Lin Chase's visERA.input.file.spec:

```
Error Region Analysis (ERA) program input file format.
Lin Chase
Carnegie Mellon University
20 July 1995

-----
<int num_utterances> UTTERANCES
UTT <utterance_id_tag1>
REF
<int num_segmentsR> SEGMENTS
<string word1R>
<int start_frame1R>
<int end_frame1R>
<int32 acoustic_score1R>
<int32 language_score1R>
<string language_score_source1R>
<string word2R>
.
.
.
HYP
<int num_segmentsH> SEGMENTS
<string word1H>
<int start_frame1H>
<int end_frame1H>
<int32 acoustic_score1H>
<int32 language_score1H>
<string language_score_source1H>
<string word2H>
.
.
.
UTT <utterance_id_tag2>
REF
.
.
.
HYP
.
.
```

```
.
EOF
-------------
```

Notes:

1. The integer in front of the token "UTTERANCES" indicates how many "UTT", "REF" and "HYP" entries there will be in the file.
2. For each REF the integer in front of the token "SEGMENTS" indicates the number of word segmentations that should be included before the next instance of the "HYP" token is encountered.
3. For each HYP the integer in front of the token "SEGMENTS" indicates the number of word segmentations that should be included before the next instance of the "REF" token is encountered.
4. "language_score_source" strings can be used to indicate algorithmic origins of language model scores, such as the branch of the Katz backoff algorithm used. The blank string "" should be used if you’d like to skip this bit.
5. The start frames of the REF and HYP sequences must be the same. The end frames of the REF and HYP sequences must be the same. The start frame of one segment within a REF/HYP sequence must be one integer count greater than the end frame of the previous segment in the sequence.

B CTM File Format

NAME

ctm - Definition of time marked conversation scoring input

DESCRIPTION

This describes the time marked conversation input files to be used for scoring the output of speech recognizers via the NIST sclite() program. Both the reference and hypothesis input files can share this format.

The ctm file format is a concatenation of time mark records for each word in each channel of a waveform. The records are separated with a newline. Each word token must have a waveform id, channel identifier [A | B], start time, dura- tion, and word text. Optionally a confidence score can be appended for each word. Each record follows this BNF for- mat:

```
CTM :== <F> <C> <BT> <DUR> word [ <CONF> ]
       Where :
         <F>  ->
              The waveform filename.
              NOTE: no pathnames or
              extensions are expected.
         <C>  ->
              The waveform channel.
              Either "A" or "B".  The text
              of the waveform channel is
              not restricted by sclite.
              The text can be any text
              string without witespace so
              long as the matching string
              is found in both the refer-
              ence and hypothesis input
              files.
         <BT> ->
              The begin time (seconds) of
              the word, measured from
              the start time of the file.
         <DUR>  ->
```

The duration (seconds) of the word.

<CONF> ->

Optional confidence score. It is proposed that this score will be used in the future.

The file must be sorted by the first three columns: the first and the second in ASCII order, and the third by a numeric order. The UNIX sort command: "sort +0 -1 +1 -2 +2nb -3" will sort the words into appropriate order.

Lines beginning with ';;' are considered comments and are ignored. Blank lines are also ignored.

Included below is an example:

```
;;
;;  Comments follow ';;'
;;
;;  The Blank lines are ignored
;;

7654 A 11.34 0.2  YES -6.763
7654 A 12.00 0.34 YOU -12.384530
7654 A 13.30 0.5  CAN 2.806418
7654 A 17.50 0.2  AS 0.537922
:
7654 B 1.34 0.2  I -6.763
7654 B 2.00 0.34 CAN -12.384530
7654 B 3.40 0.5  ADD 2.806418
7654 B 7.00 0.2  AS 0.537922
:
```

For CTM reference files, a format extension exists to permit marking alternate transcripts. The alternation uses the same file format as described above, except three word strings, "<ALT_BEGIN>", "<ALT>" and "<ALT_END>", are used to delimit the alternation. Each tag is treated as a word, with a conversation id, channel and "*"'s for the begin and duration time.

The alternation is begun using the word "<ALT_BEGIN>", and terminated using the word "<ALT_END>". In between the start and end, are at least 2 alternative time-marked word sequences separated by the word "<ALT>". Each word sequence can contain any number of words. An empty alternative sig- nifies a null word.

Below is and example alternate reference transcript for the words "uh" and "um".

```
;;
7654 A   *    *   <ALT_BEGIN>
7654 A 12.00 0.34 UM
7654 A   *    *   <ALT>
7654 A 12.00 0.34 UH
7654 A   *    *   <ALT_END>
```

SEE ALSO

sclite(1)