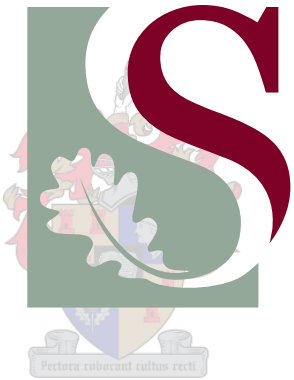
Non-Acoustic Speaker Recognition

Ilze du Toit



Thesis presented in partial fulfilment of the requirements for the degree
Master of Science in Electronic Engineering
at the University of Stellenbosch

Supervisor: Prof. J.A. du Preez

December 2004

Declaration

I, the undersigned, hereby declare that the work contained in this thesis is my own original work and that I have not previously in its entirety or in part submitted it at any university for a degree.

Signature

Date

Abstract

In this study the phoneme labels derived from a phoneme recogniser are used for phonetic speaker recognition. The time-dependencies among phonemes are modelled by using *hidden Markov models* (HMMs) for the speaker models. Experiments are done using first-order and second-order HMMs and various smoothing techniques are examined to address the problem of data scarcity. The use of word labels for lexical speaker recognition is also investigated. Single word frequencies are counted and the use of various word selections as feature sets are investigated. During April 2004, the University of Stellenbosch, in collaboration with Spescom DataVoice, participated in an international speaker verification competition presented by the *National Institute of Standards and Technology* (NIST). The University of Stellenbosch submitted phonetic and lexical (non-acoustic) speaker recognition systems and a fused system (the primary system) that fuses the acoustic system of Spescom DataVoice with the non-acoustic systems of the University of Stellenbosch. The results were evaluated by means of a cost model. Based on the cost model, the primary system obtained second and third position in the two categories that were submitted.

Oorsig

Hierdie projek maak gebruik van foneem-etikette wat geklassifiseer word deur 'n foneemherkenner en daarna gebruik word vir fonetiese sprekerherkenning. Die tyd-afhanklikhede tussen foneme word gemodelleer deur gebruik te maak van *verskuilde Markov modelle* (HMMs) as sprekermodelle. Daar word geëksperimenteer met eerste-orde en tweede-orde HMMs en verskeie vergladdingstegnieke word ondersoek om dataskaarsheid aan te spreek. Die gebruik van woord-etikette vir sprekerherkenning word ook ondersoek. Enkelwoordfrekwensies word getel en daar word geëksperimenteer met verskeie woordseleksies as kenmerke vir sprekerherkenning. Gedurende April 2004 het die Universiteit van Stellenbosch in samewerking met Spescom DataVoice deelgeneem aan 'n internasionale sprekerverifikasie kompetisie wat deur die *National Institute of Standards and Technology* (NIST) aangebied is. Die Universiteit van Stellenbosch het ingeskryf vir 'n fonetiese en 'n woordgebaseerde (nie-akoestiese) sprekerherkenningstelsel, asook 'n saamgesmelte stelsel wat as primêre stelsel dien. Die saamgesmelte stelsel is 'n kombinasie van Spescom DataVoice se akoestiese stelsel en die twee nie-akoestiese stelsels van die Universiteit van Stellenbosch. Die resultate is geëvalueer deur gebruik te maak van 'n koste-model. Op grond van die koste-model het die primêre stelsel tweede en derde plek behaal in die twee kategorieë waaraan deelgeneem is.

Acknowledgements

I would like to thank the following people for their help during the course of this study:

- Special thanks to my promotor, Prof. J.A. du Preez, for his aid, guidance, enthusiasm and support.

- Andre du Toit for his design of a phoneme recogniser during the 2002 speaker recognition evaluation held by the *National Institute of Standards and Technology* (NIST).

- All the people involved in the NIST 2004 speaker recognition evaluation : Niko Brümmer, Herman Engelbrecht and Francois Cilliers.

- Special thanks to my mother, Cynthia du Toit, for all the time she spent to edit this report. Thank you also to Emli-Mari Nel and my promotor, Prof. J.A. du Preez, for proofreading this report.

- Thank you to my family and friends, and especially to Ludwig Schwardt, for supporting me during the time of my tribulation.

- Gert-Jan van Rooyen for the use of his LaTeX report template.

- The various people that have contributed to the PatRec system.

Contents

1 Introduction **1**

1.1 Motivation 1

1.2 Objectives 2

1.3 Contributions 2

1.4 Overview 5

2 Speaker Recognition: Theoretical Background **6**

2.1 Basic Steps of Statistical Pattern Recognition 6

2.1.1 Creating/Training the Model 6

2.1.2 Evaluation 7

2.2 Speaker Recognition 9

2.2.1 Gaussian Mixture Model 9

2.2.2 Hidden Markov Model (HMM) 10

2.3 Verification 12

2.3.1 T-Norm Verifier 12

2.3.2 Detection Error Trade-off (DET) Curves 13

2.4 Literature Study 14

2.4.1 Acoustic 14

2.4.2 Non-Acoustic 18

3 Modelling Recogniser Errors 21

3.1 Introduction . . . 21

3.2 Our Approach . . . 21

3.3 Databases . . . 22

3.3.1 TIMIT Database . . . 22

3.3.2 NTIMIT Database . . . 23

3.3.3 1996 ICSI Transcriptions . . . 23

3.3.4 Switchboard I and II Corpus . . . 24

3.4 Phoneme Modelling . . . 25

3.4.1 Feature Extraction . . . 25

3.4.2 Model Structure . . . 25

3.4.3 Substitution, Insertion, Deletion (SID) Counts . . . 25

3.4.4 Incorporating the Databases with the Phoneme Modelling . . . 27

3.5 Speaker Modelling . . . 27

3.5.1 Universal Background Model . . . 27

3.5.2 Training and Evaluation Setup . . . 27

3.5.3 Model Structure . . . 29

3.5.4 Initialisation and Training . . . 30

3.5.5 Verification . . . 31

3.6 Experiments . . . 32

3.6.1 Modelling of Substitutions vs no Modelling of Substitutions in the PDFs . . . 32

3.6.2 First-Order vs Second-Order HMMs . . . 34

3.6.3 Initialisation of Transition Probabilities . . . 35

3.7 Summary . . . 37

4 Addressing the Problem of Data Scarcity 38

4.1 Introduction . . . 38

4.1.1 Chapter Outlay . . . 38

4.2 Addressing the Data Scarcity Problem . . . 39

4.2.1 Using Fewer Parameters . . . 39

4.2.2 Using Prior Models to Smooth Other Models . . . 43

4.3 Experimental Approach . . . 44

4.3.1 Merging Labels . . . 45

4.3.2 Uniform Prior Probabilities . . . 46

4.3.3 UBM Prior Models and Higher-Order Smoothing . . . 46

4.3.4 Smoothing using Merged Models as Prior Models . . . 46

4.3.5 Merging Labels in Combination with other Smoothing Techniques . 47

4.4 Experimental Results . . . 47

4.4.1 Merging Labels . . . 48

4.4.2 Smoothing with Uniform Prior Probabilities . . . 50

4.4.3 UBM Prior Models and Higher-Order Model Smoothing . . . 51

4.4.4 Merging Labels in Combination with other Smoothing Techniques . 55

4.4.5 Summary . . . 55

5 Significance Tests for Second-Order Experiments 57

5.1 Introduction . . . 57

5.2 Significance Tests . . . 58

5.3 Experimental Approach . . . 59

5.3.1 Newly Defined Dirichlet Estimator . . . 59

5.3.2 Using the McNemar Test with DET curves . . . 60

5.3.3 Experimental Setup . . . 61

5.4 Experiments using the McNemar Test and DET curves . . . 61

5.5 Summary . . . 62

6 A lexical approach to Speaker Recognition 64

6.1 Introduction . . . 64

6.2 Background . . . 65

6.3 Speaker Modelling . . . 65

6.3.1 Databases and Handset Labels . . . 65

6.3.2 Model Structure . . . 66

6.4 Selection of Word Labels as Feature Set . . . 67

6.4.1 Selection based upon word count . . . 67

6.4.2 Selection of Words Based upon Speaker Entropy . . . 68

6.4.3 Selection of Words Based upon Log-Probabilities . . . 68

6.5 Training Genuine Word Labels : Approach . . . 69

6.6 Experiments . . . 71

6.6.1 Selection of Words Based upon Minimum Word Counts . . . 71

6.6.2 Selection of Words Based upon Entropy and Log-Probabilities . . . 73

6.6.3 Training Genuine Word Labels . . . 73

6.7 Summary . . . 75

7 Combining Verifiers **77**

7.1 Introduction . . . 77

7.2 The verifier scores used for combination . . . 78

7.3 UBM, Target, Impostor and Validation Sets . . . 79

7.4 Combining the verifier output scores . . . 80

7.4.1 Verifier Selection and Verifier averaging . . . 80

7.4.2 Treating the output scores as inputs to a second-level verification problem . . . 81

7.5 Experiments and results . . . 82

7.6 Conclusion . . . 84

8 NIST 2004 Evaluation **85**

8.1 Introduction . . . 85

8.2 Task Definition and Evaluation Conditions . . . 86

8.3 Development Data . . . 86

8.4 Evaluation Data . . . 87

8.5 System Description . . . 88

8.5.1 SDV_0 . . . 88

8.5.2 SDV_2 . . . 88

8.5.3 SDV_3 . . . 89

8.5.4 System SDV_4 . . . 90

8.6 Choosing the Decision Threshold . . . 90

8.7 Computational Statistics . . . 91

8.8 Results . . . 91

8.8.1 Overall Performance Including All Trials . . . 91

8.8.2 English Language Single Handset (ELSH) 94
8.8.3 Same and Different Language Target Trials 97
8.8.4 Same and Different Language Non-Target Trials 99
8.8.5 The Influence of Cellular and Cordless Phones 101
8.8.6 Comparison of Evaluation Results with Development Results 103
8.9 Conclusion 104

9 Conclusion **106**

9.1 Introduction 106
9.2 Phonetic Speaker Recognition 106
9.2.1 The Phoneme Recogniser System 106
9.2.2 Speaker Model Structure 107
9.2.3 Smoothing of Speaker Models 107
9.3 Lexical Speaker Recognition 107
9.4 Verifier Combination 108
9.5 NIST 2004 Evaluation 108
9.6 Recommendations 109
9.7 Final Conclusion 110

A Phonetic Speaker Recognition **116**

A.1 Modelling Phoneme Recogniser Errors 116

A.2 First and Second-Order Experiments without Smoothing 117

A.3 Merging Labels 120

A.4 Experiments using Merged Labels 123

A.5 Merged Models as Prior Models 124

A.5.1 Concept and Approach 124

A.5.2 Experiments 129

A.6 Experiments Using Merged Models in Combination with Other Smoothing Techniques 133

A.6.1 Combined with Uniform Smoothing 133

A.6.2 Combined with Dirichlet Smoothing 134

A.7 Observation Counts 134

A.8 Significant Probability Levels and DET curves 134

A.8.1 First-Order Experiments with and without Smoothing 134

A.8.2 First-Order and Second-Order Experiments 137

B Lexical Speaker Recognition **140**

B.1 Influence of the Minimum word count 140

B.2 Influence of the Maximum word count 142

B.3 Selection of words based upon Entropy 146

B.3.1 Feature sets 147

B.4 Selection of words based upon Log-probabilities 151

B.5 Training Genuine Word Labels 154

C Verifier Combination Techniques **157**

C.1 DET Curves 157

C.2 Significant Probability levels of Fusion Techniques 158

D NIST 2004 Evaluation **164**

D.1 Computational statistics 164

D.2 DET Curves 166

D.2.1 8 Conversation Sides - English Language Single Handset (ELSH) . . 166

D.2.2 16 Conversation Sides - English Language Single Handset 169

D.2.3 8 Conversation Sides - Same and Different Language Target Trials . 173

D.2.4 8 Conversation Sides - Same and Different Language Non-Target Trials 176

D.2.5 16 Conversation Sides - Same and Different Language Target Trials 179

D.2.6 16 Conversation Sides - Same and Different Language Non-Target Trials 183

D.2.7 Phone Types 187

List of Figures

2.1 A block-diagram illustrating the basic steps of pattern recognition for a verification task. 8

2.2 A typical representation of a Hidden Markov Model with the output probability density functions. 11

3.1 The phoneme recogniser system 22

3.2 Basic representation of the initialisation and training process followed for first-order HMM speakers and impostors. 31

3.3 DET curves for modelling of substitution errors and no modelling thereof, using 4 conversation sides for training. 33

3.4 DET curves for first-order, X1(EP), and second-order experiments, X2(EP), using 16 conversation sides for training. 35

3.5 DET curves comparing initialisation of EP and BG of first-order experiments. 36

3.6 DET curves comparing initialisation of EP and BG of second-order experiments. 36

4.1 A matrix representation of substitution counts before and after label merging of a common label with a common label takes place. 42

4.2 A flowchart of the system describing the merging of labels, the computation of substitution counts and the generation of new merged feature sets. . . . 43

4.3 Example of possible paths for UBM prior models to smooth target speaker models (TSMs) and higher-order smoothing up to third-order HMMs. . . . 47

4.4 DET curves of experiments using first-order TSMs as prior models to Dirichlet smooth second-order TSMs compared to experiments where smoothing is done using uniform prior probabilities. 54

5.1 DET curve illustrating areas of significant difference of X1 and X2 using 16 conversation sides for training. 62

6.1 The calculation of (a) substitution probabilities and (b) reversed substitution probabilities. 70

6.2 DET curves of feature sets containing words that are selected based on minimum word counts, using 4 conversation sides for training. 72

6.3 Comparison of DET curves of experiments that trained genuine labels (TGL) and experiments that trained with classified labels (TCL) using 4 conversation sides and feature set FEAT1. 75

7.1 DET curves comparing the different verifier combination techniques with their constituent verifiers, using 16 conversation sides for training. 83

8.1 DET curves of all the systems including all trials, using 8 (5 minute) conversation sides for training and 1 conversation side for test segment trials. 92

8.2 DET curves of all the systems including all trials, using 16 (5 minute) conversation sides for training and 1 conversation side for test segment trials. 92

8.3 DET curves (pooled gender, male and female trials) of SDV_4 (ELSH data), using 8 (5 minute) conversation sides for training and 1 conversation side for test segment trials. 95

8.4 DET curves of same and different language target trials of SDV_4, using 8 (5 minute) conversation sides for training and 1 conversation side for test segment trials. 98

8.5 DET curves of same and different language non-target trials of SDV_4, using 8 (5 minute) conversation sides for training and 1 conversation side for test segment trials. 100

8.6 DET curves comparing trials where only regular phones are used in comparison to the overall performance where any type of phone is used (including all trials) of SDV_3, using 8 (5 minute) conversation sides (CS) for training. 102

8.7 DET curves showing the performance of trials where models are trained using 8 conversation sides and **cellular phones**. 103

8.8 DET curves of all the systems using the data of jackknife set 0 (jack_0) from Switchboard II. All trials of speakers trained using 8 and 16 (3 minute) conversation sides are used. 104

A.1 DET curves for modelling of substitution errors and no modelling thereof, using 8 conversation sides for training. 116

A.2 DET curves for modelling of substitution errors and no modelling thereof, using 16 conversation sides for training. 117

A.3 DET curves for first-order, X1(EP), and second-order experiments, X2(EP), using 4 conversation sides for training. 117

A.4 DET curves for first-order, X1(EP), and second-order experiments, X2(EP), using 8 conversation sides for training. 118

A.5 DET curves for first-order, X1(BG), and second-order experiments, X2(BG), using 4 conversation sides for training. 118

A.6 DET curves for first-order, X1(BG), and second-order experiments, X2(BG), using 8 conversation sides for training. 119

A.7 DET curves for first-order, X1(BG), and second-order experiments, X2(BG), using 16 conversation sides for training. 119

A.8 A matrix representation of substitution counts, $\mathbf{C} = c_{ij}$ before label merging takes place. 121

A.9 The equivalent unmerged model (right) of a merged model (left), with state 2 being a state with merged label $l_2|l_3$, merged from labels l_2 and l_3. 125

A.10 A representation of the fusion and training of unmerged equivalent models. The dashed arrows illustrate the computation of equivalent unmerged models. 125

A.11 A compact simplification, replacing the fusion and training structure of that of Figure A.10. 126

A.12 The structure of experiments taking Route A described in Section A.5.1 . 127

A.13 An illustration of the first step of the structure of Route B described in Section A.5.1. 128

A.14 An illustration of the second step of the structure of Route B described in Section A.5.1 128

A.15 The total structure of Route B. 129

A.16 Significant probability levels plotted against $r = FRR/FAR$ where X_A performs better than X_B, using 4 conversation sides for training. 135

A.17 Significant probability levels plotted against $r = FRR/FAR$ where X_A performs better than X_B, using 8 conversation sides for training. 136

A.18 Significant probability levels plotted against $r = FRR/FAR$ where X_A performs better than X_B, using 16 conversation sides for training. 136

A.19 DET curve illustrating areas of significant difference of X1 and X2 using 4 conversation sides for training. 137

A.20 DET curve illustrating areas of significant difference of X1 and X2 using 8 conversation sides for training. 138

A.21 Significance Probability P vs $r = FRR/FAR$, where X2 performs better than X1, using 4 conversation sides for training. 138

A.22 Significance Probability P vs $r = FRR/FAR$, where X2 performs better than X1, using 8 conversation sides for training. 139

A.23 Significance Probability P vs $r = FRR/FAR$, where X2 performs better than X1, using 16 conversation sides for training. 139

B.1 DET curves of feature sets containing words that are selected based on minimum word counts, using 8 conversation sides for training. 140

B.2 DET curves of feature sets containing words that are selected based on minimum word counts, using 16 conversation sides for training. 141

B.3 DET curves of feature sets excluding words with the highest word counts, using 4 conversation sides for training. 143

B.4 DET curves of feature sets excluding words with the highest word counts, using 8 conversation sides for training. 144

B.5 DET curves of feature sets excluding words with the highest word counts, using 16 conversation sides for training. 145

B.6 DET curves of experiments using a feature set containing 4000 unique word labels: One generated using the speaker entropy of 641 UBM speakers and the other using the 4000 words with the highest word count out of the same set of speakers. 149

B.7 DET curves of experiments using a feature set containing 4000 unique word labels: One generated using the speaker entropy of 641 UBM speakers and the other using the 6000 words with the highest word count out of the same set of speakers. 149

B.8 DET curves of experiments using a feature set containing 2000 unique word labels: One generated using the speaker entropy of 31 target speakers and the other using the 2000 words with the highest word count out of the same set of speakers. 150

B.9 DET curves of experiments using a feature set containing 3000 unique word labels: One generated using the speaker entropy of 31 target speakers and the other using the 3000 words with the highest word count out of the same set of speakers. 150

B.10 DET curves of experiments using feature sets (with 2 774 words) computed from log-probabilities using the different conversation sides (CS). LP_I uses no prioring and LP_II uses UBM prioring. 153

B.11 Comparison of DET curves of experiments that trained genuine labels (TGL) and experiments that trained with classified labels (TCL) using 8 conversation sides and feature set FEAT1. 154

B.12 Comparison of DET curves of experiments that trained genuine labels (TGL) and experiments that trained with classified labels (TCL) using 16 conversation sides and feature set FEAT1. 155

B.13 Comparison of DET curves of experiments that trained genuine labels (TGL) and experiments that trained with classified labels (TCL) using 4 conversation sides and feature set FEAT2. 155

B.14 Comparison of DET curves of experiments that trained genuine labels (TGL) and experiments that trained with classified labels (TCL) using 8 conversation sides and feature set FEAT2. 156

B.15 Comparison of DET curves of experiments that trained genuine labels (TGL) and experiments that trained with classified labels (TCL) using 16 conversation sides and feature set FEAT2. 156

C.1 DET curves comparing the different verifier combination techniques with their constituent verifiers, using 4 conversation sides for training. 157

C.2 DET curves comparing the different verifier combination techniques with their constituent verifiers, using 8 conversation sides for training. 158

C.3 Significant Probability levels of "Fuse_Gaussian" performing better than verifier C vs $r = FRR/FAR$, using 4 conversation sides for training. . . . 159

C.4 Significant Probability levels of "Fuse_Gaussian" performing better than verifier C vs $r = FRR/FAR$, using 8 conversation sides for training. . . . 159

C.5 Significant Probability levels of "Fuse_Gaussian" performing better than verifier C vs $r = FRR/FAR$, using 16 conversation sides for training. . . . 160

C.6 Significant Probability levels of "Fuse_GMM" performing better than verifier C vs $r = FRR/FAR$, using 4 conversation sides for training. 160

C.7 Significant Probability levels of "Fuse_GMM" performing better than verifier C vs $r = FRR/FAR$, using 8 conversation sides for training. 161

C.8 Significant Probability levels of "Fuse_GMM" performing better than verifier C vs $r = FRR/FAR$, using 16 conversation sides for training. 161

C.9 Significant Probability levels of "S&F" performing better than verifier C vs $r = FRR/FAR$, using 4 conversation sides for training. 162

C.10 Significant Probability levels of "S&F" performing better than verifier C vs $r = FRR/FAR$, using 8 conversation sides for training. 162

C.11 Significant Probability levels of "S&F" performing better than verifier C vs $r = FRR/FAR$, using 16 conversation sides for training. 163

D.1 DET curves (pooled gender, male and female trials) of SDV_0 (ELSH data), using 8 (5 minute) conversation sides for training and 1 conversation side for test segment trials. 166

D.2 DET curves (pooled gender, male and female trials) of SDV_2 (ELSH data), using 8 (5 minute) conversation sides for training and 1 conversation side for test segment trials. 167

D.3 DET curves (pooled gender, male and female trials) of SDV_3 (ELSH data), using 8 (5 minute) conversation sides for training and 1 conversation side for test segment trials. 168

D.4 DET curves (pooled gender, male and female trials) of SDV_0 (ELSH data), using 16 (5 minute) conversation sides for training and 1 conversation side for test segment trials. 169

D.5 DET curves (pooled gender, male and female trials) of SDV_2 (ELSH data), using 16 (5 minute) conversation sides for training and 1 conversation side for test segment trials. 170

D.6 DET curves (pooled gender, male and female trials) of SDV_3 (ELSH data), using 16 (5 minute) conversation sides for training and 1 conversation side for test segment trials. 171

D.7 DET curves (pooled gender, male and female trials) of SDV_4 (ELSH data), using 16 (5 minute) conversation sides for training and 1 conversation side for test segment trials. 172

D.8 DET curves of same and different language target trials of SDV_0, using 8 (5 minute) conversation sides for training and 1 conversation side for test segment trials. 173

D.9 DET curves of same and different language target trials of SDV_2, using 8 (5 minute) conversation sides for training and 1 conversation side for test segment trials. 174

D.10 DET curves of same and different language target trials of SDV_3, using 8 (5 minute) conversation sides for training and 1 conversation side for test segment trials. 175

D.11 DET curves of same and different language non-target trials of SDV_0, using 8 (5 minute) conversation sides for training and 1 conversation side for test segment trials. 176

D.12 DET curves of same and different language non-target trials of SDV_2, using 8 (5 minute) conversation sides for training and 1 conversation side for test segment trials. 177

D.13 DET curves of same and different language non-target trials of SDV_3, using 8 (5 minute) conversation sides for training and 1 conversation side for test segment trials. 178

D.14 DET curves of same and different language target trials of SDV_0, using 16 (5 minute) conversation sides for training and 1 conversation side for test segment trials. 179

D.15 DET curves of same and different language target trials of SDV_2, using 16 (5 minute) conversation sides for training and 1 conversation side for test segment trials. 180

D.16 DET curves of same and different language target trials of SDV_3, using 16 (5 minute) conversation sides for training and 1 conversation side for test segment trials. 181

D.17 DET curves of same and different language target trials of SDV_4, using 16 (5 minute) conversation sides for training and 1 conversation side for test segment trials. 182

D.18 DET curves of same and different language non-target trials of SDV_0, using 16 (5 minute) conversation sides for training and 1 conversation side for test segment trials. 183

D.19 DET curves of same and different language non-target trials of SDV_2, using 16 (5 minute) conversation sides for training and 1 conversation side for test segment trials. 184

D.20 DET curves of same and different language non-target trials of SDV_3, using 16 (5 minute) conversation sides for training and 1 conversation side for test segment trials. 185

D.21 DET curves of same and different language non-target trials of SDV_4, using 16 (5 minute) conversation sides for training and 1 conversation side for test segment trials. 186

D.22 DET curves showing the performance of trials where models are trained using 8 conversation sides and **cordless phones**. 187

List of Tables

2.1 Some results of speaker recognition between 1991 and 1997. 17

3.1 NIST extended data description for Switchboard II, version 2 29

3.2 EERs of modelling of SEP vs no modelling of SEP. 33

4.1 EERs of first-order results X1(BG). 48

4.2 EERs of second-order results X2(BG). 49

4.3 Smoothing first-order target models with uniform prior probabilities (X1_UP) vs no smoothing of first-order target models (X1) 51

4.4 Smoothing second-order target models with uniform prior probabilities (X2_UP) vs no smoothing of second-order target models (X2) 51

4.5 The EERs of experiments using the UBM as prior model for smoothing TSMs. 52

4.6 The EERs of experiments using the first-order target models as to smooth second-order target models. 53

4.7 Summary of the ideas that surfaced exploring several configurations of smoothing models and merging phoneme labels 55

6.1 EER results of feature sets selecting words based on minimum word counts. 72

7.1 Comparison of EERs of various fusion techniques. 84

8.1 A summary of the NIST 2004 evaluation data. 87

8.2 Actual C_{DET} costs for the various systems including all trials. 93

8.3 Approximate EERs of English Language Single Handset data. 94

8.4 Actual C_{DET} costs for the various systems for ELSH data. 96

8.5 Approximate EERs of Same/Different Language Target trials. 99

8.6 Approximate EERs of Same/Different Language Non-target trials. 101

A.1 EERs of first-order results X1(EP). 123

A.2 EERs of second-order results X2(EP). 123

A.3 The EER results using the transition probabilities of **equivalent unmerged models** as priors, with a Dirichlet prior factor $\alpha = 20$ and a fusion probability of $\beta = 0.1$. 130

A.4 The EER results using the transition probabilities of **equivalent unmerged models** as priors, with a Dirichlet prior factor $\alpha = 20$ and a fusion probability of $\beta = 0.1$. 130

A.5 The EER results using the transition probabilities of **equivalent unmerged models** as priors, with a Dirichlet prior factor $\alpha = 20$ and a fusion probability of $\beta = 0.3$. 131

A.6 The EER results using the transition probabilities of **equivalent unmerged models** as priors, with a Dirichlet prior factor $\alpha = 20$ and a fusion probability of $\beta = 0.3$. 131

A.7 The EER results using the transition probabilities of **equivalent unmerged models** as priors, with a Dirichlet prior factor $\alpha = 20$ and a fusion probability of $\beta = 0.2$. 132

A.8 The EERs of experiments using the merging of labels in combination with using **Uniform Priors, smoothing first-order** HMM speaker models (X1_UP). 133

A.9 The EERs of experiments using the merging of labels in combination with using **Uniform Priors, smoothing second-order** HMM speaker models (X2_UP). 133

A.10 The EERs of experiments using the merging of labels in combination with using first-order TSMs as prior models for smoothing second-order TSMs. . 134

A.11 Typical ranges of the total number of observation counts in a state of a speaker HMM, using NIST Switchboard II. 134

B.1 EER results of feature sets excluding words with the highest word counts. . 142

B.2 EERs using the feature sets described in section B.3 of which the DET curves are shown in Figures B.6 to B.9. 148

B.3 The results when using target-specific feature sets ascociated with each target model, generated by evaluating log-probabilities of the UBM and target models. 154

Glossary

Bigram - See N-gram.

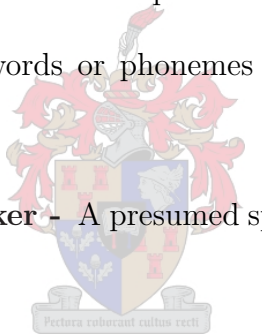
Bigram count - The number of times a given bigram appears in speech.

Classified phoneme (label) - The phoneme that was recognised by a phoneme recogniser from a part of speech.

Conversation side - A single channel side of a given speaker from a specific conversation.

Genuine phoneme (label) - The actual phoneme uttered by a speaker.

Idiosyncrasies - The use of words or phonemes of a person that is peculiar to that person.



Impostor (non-target) speaker - A presumed speaker of a test segment who is *in fact not* the actual speaker.

Impostor trial - A trial in which the actual speaker of the test segment is *in fact not* the presumed speaker.

Jackknifing procedure - A procedure that rotates through the training and test data in order to provide an adequate number of tests.

Lexical speaker recognition - Speaker recognition using word labels as features.

N-gram - Approximate word (phoneme) history by most recent $N-1$ words (phonemes). When $N = 2$ it is referred to as a bigram and when $N = 3$ it is referred to as a trigram.

Phonetic speaker recognition - Speaker recognition using phoneme labels as features.

Prior model - Model used to smooth other models.

Prior probabilities - probabilities used for smoothing.

Speaker identification - identifying a speaker out of a group of speakers.

Speaker verification - checking whether the speaker is whom he/she is claimed to be.

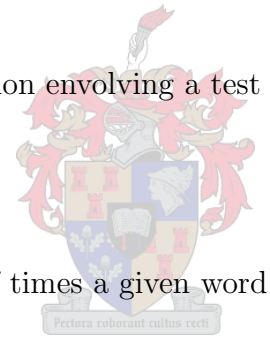
Target speaker - The presumed speaker of a test segment, for whom a model has been created from training data.

Target trial - A trial in which the actual speaker of the test segment is *in fact* the presumed speaker.

Trial - The individual evaluation envolving a test segment and a target model.

Trigram - See N-gram.

Word count - The number of times a given word appears in speech.



Acronyms

BG	- Initialisation of transition probabilities using Bigram counts.
CS	- Conversation side
DCF	- Detection Cost Function
DET curve	- Detection Error Trade-off curve
DTW	- Dynamic time warping
EER	- Equal error rate
EP	- Initialisation of transition probabilities using Equal Probabilities
FA	- False Acception or False Alarm (the test segment from an impostor speaker is classified incorrectly as a target segment.)
FR	- False Rejection (or False Miss) (the test segment from the target speaker is classified incorrectly as an impostor segment.)
FAR	- False Acceptance Ratio
FRR	- False Rejection Ratio
GMM(s)	- Gaussian Mixture Model(s)
HMM(s)	- Hidden Markov Model(s)
MAP	- *Maximum a posteriori*
MFCC(s)	- Mel-Frequency Cepstra Coefficient(s)
NIST	- National Institute of Standards and Technology
PDF(s)	- Probability Density Function(s)
UBM(s)	- Universal Background Model(s) used to assist in the training of target models.
TIMIT	- Corpus of speech collected at *Texas Instruments* (TI) and Massachusetts Institute of Technology (MIT).
TSM(s)	- Target Speaker Model(s)

Notation

α	Prior factor
$\mathbf{A} = [a_{ij}]$	state transition probability matrix
$\mathbf{C} = [c_{ij}]$	Matrix of substitution counts, where i is associated with classified labels and j with genuine labels.
C_{FA}	False alarm cost
C_{FR}	False rejection cost
$f(.)$	Probability density function (PDF)
$f(x\|j)$	Probability density associated with state j
$P(.)$	Probability
$P(CL\|GL)$	Substitution error probabilities (SEP) of classified labels (CL) and genuine labels (GL).
$P(FA\|Impostor)$	False acceptance ratio (FAR), the probability of false acceptance given that the actual speaker is a impostor speaker.
$P(FR\|Target)$	False rejection ratio (FRR), the probability of false rejection given that the actual speaker is a target speaker.
$P(GL\|CL)$	Reversed substitution error probabilities of genuine labels (GL) and classified labels (CL).
$\mathbf{PG}n$	Phoneme group with n labels in the feature set.
$X1$	First-order model
$X2$	Second-order model

Chapter 1

Introduction

1.1 Motivation

Speaker recognition falls under the general task of pattern recognition of which there are two main tasks: speaker verification and speaker identification. The goal of verification is to determine from a test voice sample whether the speaker is whom he/she is claimed to be. In identification, the goal is to determine which one of a known group of speakers best matches the test voice sample. Speaker identification can be subdivided into two main categories: closed-set and open-set. With the closed-set task, a speaker is identified from a group of N known speakers. With the open-set task, the options are extended by also allowing that the speaker be identified as unknown to the system.

Traditionally, text-independent speaker recognition is done by choosing a set of acoustic parameters, such as cepstral features, and by using Gaussian mixture speaker models or multi-layer perceptrons [45, 18, 9]. This type of speaker recognition is referred to as acoustic speaker recognition. Acoustic speaker recognition focuses on spectral differences, and the physical aspects, such as the vocal tract, are investigated.

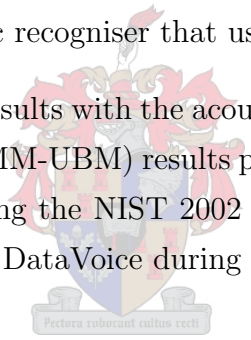
This study, however, focuses on non-acoustic speaker recognition. The focus is not on how the speech moves through the vocal tract, but rather on the usage of certain words, phrases or phonemes that is peculiar to a speaker, i.e. idiosyncrasies. A person's idiosyncrasies are influenced by his/her social environment, such as family and friends, or could be individual habits picked up with time. These idiosyncrasies are recognisable by the human listener and are the reason why humans distinguish among speakers who are familiar to them far better than those who are not. This is the reason for the moderately new interest in employing such idiosyncrasies in statistical speaker recognition, particularly by using

phonetic features or word unigrams [11, 25, 1]. Non-acoustic speaker recognition relies on longer speech patterns than what acoustic speaker recognition does. Because of the different focuses of the two speaker recognitions, non-acoustic speaker recognition can contribute to acoustic speaker recognition if the two are used in combination: Non-acoustic speaker recognition focuses on higher-level influences of the speaker, while acoustic speaker recognition focuses on the physical aspects of the speaker.

1.2 Objectives

The following are the objectives of this study:

1. To design and evaluate a number of configurations of a non-acoustic recogniser that employs phoneme labels.
2. To evaluate a non-acoustic recogniser that uses automatically-recognised words.
3. To fuse the non-acoustic results with the acoustic *Gaussian Mixture Model - Universal Background Model* (GMM-UBM) results provided by the *Massachusetts Institute of Technology* (MIT) during the NIST 2002 evaluation task and with the acoustic GMM results provided by DataVoice during the NIST 2004 evaluation task.



1.3 Contributions

1. Using a non-acoustic recogniser that employs phoneme labels, the following contributions are made:
   (a) Two first-order HMMs are compared: one that does not take the substitution errors of the phoneme recogniser into account in the probability density functions (PDFs), and one that does. One finds that taking the substitution errors into account is a considerable improvement compared to not doing so.
   (b) Different configurations are evaluated, such as different initialisation of transition probabilities and the use of higher-order HMMs. One finds that data scarcity poses a problem, especially for the higher-order HMMs.
   (c) The implementation and evaluation of several smoothing techniques as a solution for data scarcity are done:

i. The use of fewer, broader phoneme categories gives more data per link in the HMM. This is implemented by merging the phoneme labels that cause the most confusion. These merged labels are used as the new feature set to train speaker models with less parameters (we refer to these models as merged models).

ii. Smoothing of transition probabilities is investigated by making use of *maximum a posteriori* (MAP) estimation with Dirichlet prior probabilities. Prior probabilities are used to smooth the probabilities of a model in the process of training. This is not the general means of smoothing transition probabilities. We experiment with uniform prior probabilities and use transition probabilities of well-trained models to smooth the transition probabilities of other models.

iii. The transition probabilities of the merged models are used to smooth those of other models. The merging of the labels and the calculation of a new set of transition probabilities are time-consuming and this method of smoothing is found to be of little value for speaker recognition.

iv. The transition probabilities of the UBM are used as prior probabilities to smooth the transition probabilities of target speaker models. This is done to ensure that the speaker models are not over-fitted. First-order speaker models are used as prior models to initialise second-order speaker models. Using the UBM as prior model for the first-order speaker models gives similar results as when training them without smoothing. On the other hand, there is a marked improvement using first-order speaker models to Dirichlet smooth second-order speaker models compared to second-order models using no smoothing. There is insignificant improvement in using first-order speaker models as prior models for second-order speaker models, compared to using first-order models with no smoothing. However, should more data be available, the use of first-order speaker models as prior models for second-order speaker models would be worth re-evaluating.

2. Using a non-acoustic recogniser that employs classified word labels, the effect of certain word selections are investigated:

   (a) The number of times a word is used in a data set is referred to as the word frequency or word count. A selection of words is made of which the word count is greater than a chosen threshold. Different selections of words are made by varying this threshold (minimum word count), and the effects on speaker recognition are studied.

(b) A selection of words that is speaker-specific is explored by making use of speaker entropy and the log-probabilities of the UBM and speaker models. These selections of words do not work well because of data scarcity.

3. The verifiers of the acoustic results of MIT are combined with the T-Norm scores of the non-acoustic results using:

   (a) first-order speaker HMMs with a phonetic feature set (39 phonemes) and

   (b) word labels in Switchboard I as feature set, leaving out words that are used relatively seldom.

   We combine the verifiers by:

   (a) A combination of verifier selection and weighted averaging of verifier scores.

   (b) Treating the verifier scores as the input to another verifier and using statistical pattern recognition for verification. We use both *Gaussian Mixture Models* (GMM) and a Gaussian distribution for the training of scores.

   The second method works better than the first.

4. Stellenbosch University (SUN), in collaboration with Spescom Datavoice (SDV), participated in the NIST 2004 evaluation and submitted a lexical system, a phonetic system and a fused system (primary system) that fuses the non-acoustic GMM system of Spescom Datavoice with the two non-acoustic systems. The two categories participated in were 8sides-1side and 16sides-1side train/test conditions [1].

   In the official competition, there were 10 participants in the 8sides-1side category and 6 participants in the 16sides-1side category. The results were evaluated using a cost function. SUN and SDV's primary system obtained second and third position in the 16sides-1side and the 8sides-1side categories when different language[2] trials were included and performed best in both categories when the training and testing conversations were in English.

---

[1] The training and test conditions are explained in more detail in Section 8.2

[2] The language of the training and testing conversations differs.

1.4 Overview

Chapter 2 reviews the basic methods of statistical pattern recognition and the most popular statistical models used for speaker recognition. We also deal with the literature study of acoustic and non-acoustic speaker recognition over the past few years.

Chapter 3 is the first of three chapters that deal with the use of phonetic labels for speaker recognition. The phoneme recogniser system that generates the phoneme labels is discussed. We deal specifically with the substitution errors of the phoneme recogniser and how these errors can be utilised to improve the speaker recognition system. The model structure of the phonetic speaker HMM is discussed, and first- and second-order experiments are conducted without any Dirichlet estimation (smoothing). Problems of data scarcity are experienced when using no smoothing of the second-order models.

Chapter 4 investigates several possible smoothing techniques and configurations with the aim of addressing the problem of data scarcity. These experiments are evaluated by comparing the equal error rates (EERs), to see which of these ideas seem promising for speaker recognition purposes, and which not. Using first-order speaker models as prior model to smooth second-order speaker models seem to be the most promising techniques in Chapter 4.

Chapter 5 uses McNemar significance tests to evaluate results.

Chapter 6 deals with the use of word labels for speaker recognition. The model structure which is used when word labels are employed as feature set is discussed. Several selections of words are used for speaker recognition. Accordingly, their effects are investigated.

Chapter 7 shows how non-acoustic verifier output scores can be combined with acoustic verifier output scores to improve the acoustic results. Several means of verifier combinations are examined.

Chapter 8 deals with the NIST 2004 evaluation results of Stellenbosch University and Spescom Datavoice.

A conclusion of this study is given in **Chapter 9**.

Chapter 2

Speaker Recognition: Theoretical Background

Speaker recognition forms part of the pattern recognition field. In pattern recognition, the task is to recognise the object in use, be it an image or a speaker. To do this, we need to collect some knowledge about the object type. In the statistical field, this is done by creating statistical models for the object type. Statistical models are defined in terms of their parameters. Optimising these parameters with real world data is called training. For example: in speaker recognition, we would collect training data of different speakers and use it to create statistical models for the speakers. The statistical parameters of the trained model are used to compute likelihood scores of the evaluation set. It is important that we keep the training data and evaluation data separate [18]. There are different methods used to estimate the parameters of a model. Only the ones used in this thesis are discussed.

2.1 Basic Steps of Statistical Pattern Recognition

Different data sets for training and evaluation are chosen.

2.1.1 Creating/Training the Model

- A significant feature set for the given pattern recognition problem is chosen. For a speaker recognition task, this feature set would traditionally consist of cepstral features, energy, etc., calculated per speech signal frame length in time. The feature set for this study would be phoneme or word labels.

- Feature sequences are extracted over the whole range of training data.
- The feature sequence is used as input to a model estimator. Some estimators need a set of initial parameters and could be sensitive to these initial parameters. Good estimation of these parameters is therefore essential. The model estimator calculates a set of parameters for the chosen model. This can be an iterative process in which the model parameters are re-estimated to maximise the likelihood of the data, given the model. These parameters could, in turn, be the input to yet another estimator, such as a smoothing estimator.

A model is estimated for each of the different classes which needs to be distinguished or verified. In a speaker recognition task, the classes would consist of the set of speakers which needs to be distinguished from one another. A speaker model is trained for each of the speakers. Choosing the right model and having enough data are essential for good model estimation.

2.1.2 Evaluation

Evaluation can be divided into either a classification task or a verification task.

- Feature sequences for the evaluation data are extracted.
- In the case of classification: log-likelihood scores for the different classes over all the feature sequences are calculated as

$$score_k = \frac{1}{N}\sum_{n=1}^{N} log(P(x_n|Model_k)) \qquad k = 1, 2.., K \tag{2.1}$$

where K is the number of classes, x_n is the n-th element of feature sequence X, and N the length of the feature sequence. The normalisation with the feature sequence length, N, is done to keep the log-likelihood scores in an appropriate range by making them invariant to the length of the feature sequence. The evaluation sequence is classified as the class with the highest score. In the case of open-set classification, the evaluation sequence is classified as the class with the highest score above a chosen threshold; otherwise it is classified as none of the classes.

- In speaker verification the task is to determine whether the speaker of the test sequence is whom he or she is claimed to be (hypothesised speaker). This hypothesised speaker is referred to as the target speaker. Any other speaker than the target speaker is referred to as an **impostor** speaker. For instance, if a speaker claims to be John, then John is the target speaker. If the true speaker of the test sequence was in fact not John, then that speaker is an impostor speaker. In this study, the T-Norm verifier [4] is used to verify the target and impostor speakers. The T-Norm verifier is discussed in more detail in Section 2.3.1.

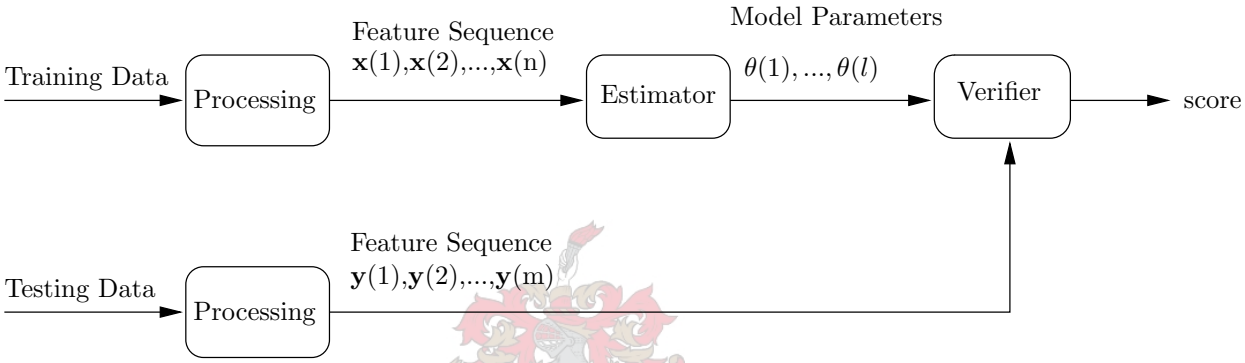


Figure 2.1: *A block-diagram illustrating the basic steps of pattern recognition for a verification task.*

Figure 2.1 illustrates the basic steps of pattern recognition for a verification task. The top half of the diagram illustrates the training process, and the bottom half illustrates the testing or verification process. The figure shows that one needs two separate data sets for training and testing. Feature sequences are extracted from both these data sets. The feature sequence of the training data is represented by $\mathbf{x}(1), \ldots, \mathbf{x}(n)$, where n is the feature sequence length. Likewise, the feature sequence of the test data is represented by $\mathbf{y}(1), \ldots, \mathbf{y}(m)$.

Statistical models with model parameters, $\theta(1), .., \theta(l)$, are estimated from training the feature sequences of the training data. Verification generates a single scalar score (generally a likelihood score). In this study, the higher the score, the more likely it is that the test sequence has been generated from the target speaker.

2.2 Speaker Recognition

Speaker recognition is a statistical pattern recognition problem. The basic steps of statistical pattern recognition can therefore be applied to speaker recognition. Typically, an acoustic feature set, such as spectral features and pitch, is modelled. The earliest approach was to use long-term averages of these acoustic features [35, 33]. Another approach is to model the speaker-dependent acoustic features within the individual phonetic sounds of the speech utterance. Acoustic features from phonetic sounds in a test utterance are compared with speaker-dependent acoustic features from similar phonetic sounds.

There are various modelling techniques, such as neural networks, uni-modal Gaussian, VQ codebook and Gaussian Mixture Models [47, 17, 51, 45]. In Section 2.2.1, the Gaussian Mixture Model (GMM) is described, since this is presently one of the most popular models used for speaker recognition.

2.2.1 Gaussian Mixture Model

In the case of a general mixture model, the model for the density can be written as a linear combination of component densities, $f(\mathbf{x}|j)$ [6]

$$f(\mathbf{x}) = \sum_{j=1}^{M} f(\mathbf{x}|j)P(j) \tag{2.2}$$

where f($\mathbf{x}$) is a density function, and P(j) is a probability of the data point being generated of component j, and should satisfy

$$\sum_{j=1}^{M} P(j) = 1 \tag{2.3}$$

where M is the number of mixtures.

The component density function $f(\mathbf{x}|j)$ per definition should satisfy

$$\int f(\mathbf{x}|j)\, d\mathbf{x} = 1 \tag{2.4}$$

A popular choice for $f(\mathbf{x}|j)$ is the multi-dimensional Gaussion PDF in which case the mixture is known as a Gaussian Mixture Model (GMM). This also is a common choice for modelling sequences of statistical feature vectors as a product of GMM PDF heights. In the case of a Gaussian model, $M = 1$.

In the case of GMMS

$$f(\mathbf{x}) = \frac{1}{(2\pi)^{d/2}|\mathbf{C}|^{1/2}} exp\left\{-\frac{1}{2}(\mathbf{x}-\mu)^T\mathbf{C}^{-1}(\mathbf{x}-\mu)\right\} \tag{2.5}$$

where μ is the d-dimensional mean vector, $\mathbf{C}$ is a $d \times d$ covariance matrix, and $|\mathbf{C}|$ is the determinant of $\mathbf{C}$. Estimation of the GMMs is done by using the *EM-algorithm* [6]. The *EM-algorithm* is iterative and sensitive to initialisation. The initial state can be computed by a binary split method, and then using the *K-means clustering algorithm* on this binary split.

2.2.2 Hidden Markov Model (HMM)

If there are time dependencies between the features, they can be modelled using an HMM. Discrete HMMs can be described by the following parameters (taken in part from [44] and [13]):

- There are a finite number of states, N, in the model.
- At each time step, t, a new state is entered, based upon a transition probability distribution which depends on the previous state. The transition may be so that the process remains in the previous state. The process can occupy only one state at a time.
- After each transition is made, an observation output symbol is produced according to a probability distribution which depends on the current state. The distribution remains fixed for the current state, regardless of how and when the state is entered. There are thus N probability distributions, corresponding to each of the N states.
- For a sequence of observed symbols, a corresponding but hidden sequence of states exists. Hence the name *hidden* Markov model.

Figure 2.2 shows a typical HMM used in speech processing. The small black dots represent non-emitting states (an initial and terminating state) that replace a separate initial state distribution of the HMM. The initial state has only outgoing transition links and the terminating state only incoming ones. These non-emitting states have no PDFs associated with them. All the emitting states are represented in the figure with circles. Each emitting state has a PDF associated with it.

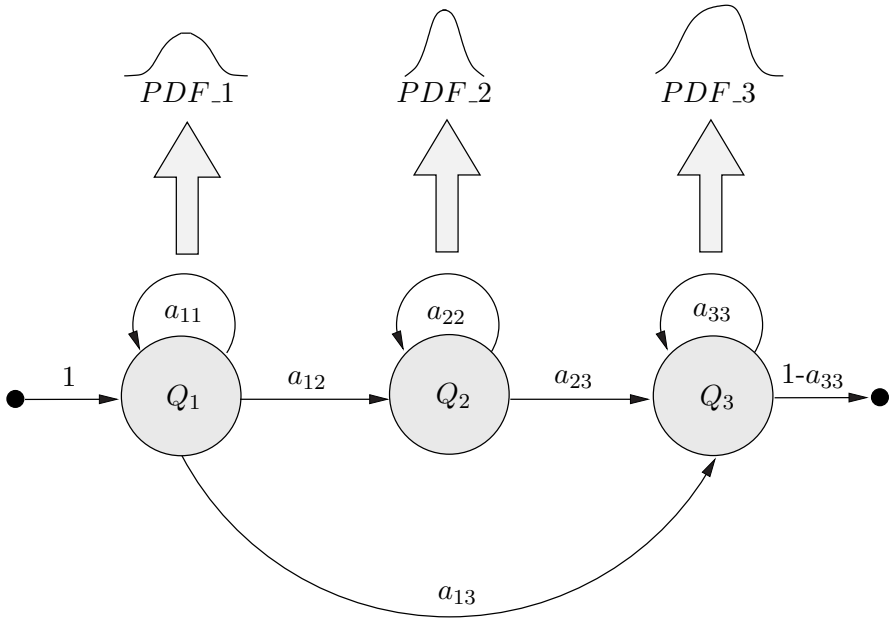


Figure 2.2: *A typical representation of a Hidden Markov Model with the output probability density functions.*

The following model notation for a first-order HMM is defined and used in Figure 2.2:

- T = length of the observation sequence (total number time steps)
- N = number of states in the model
- $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T)$, observation sequence (feature sequence)
- $\mathbf{S} = (\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_T)$, hidden state sequence
- $f(\mathbf{x}|j)$, the PDF associated with emitting state j, where $j = 1, 2, \ldots N$.
- $\mathbf{Q} = (\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_N)$, states.
- $\mathbf{A} = [a_{ij}]$, $a_{ij} = P_r(s_t = q_j | s_{t-1} = q_i)$, $i, j = 0, 1, \ldots N+1, t = 1, \ldots, T$, state transition probability matrix.

The HMM can be described by the following compact notation: $\lambda = (\mathbf{A}, f(\mathbf{x}|j))$. The Viterbi re-estimation algorithm is used to estimate the parameters of the HMM. It calculates the most likely hidden state sequence that produces the observed sequence [44]. The specific HMM structure shown in Figure 2.2 is referred to as a left-to-right HMM. In this particular structure all the transitions are from the current state to a state that

is either the same or later in time, but never earlier in time. With states that follow chronologically from left to right, the transitions have a flow mainly in the right direction, hence the name *left-to-right* HMM. An ergodic HMM has a structure with no restrictions on the direction of transitions from one state to another.

A fully connected ergodic HMM is an HMM, where every emitting state has an outgoing transition link to itself and all other emitting states. The non-emitting initial state has outgoing transition links to all the emitting states. All the emitting states have transition links to the terminating state.

2.3 Verification

2.3.1 T-Norm Verifier

There are several verifiers that can be used for speaker recognition. In this study the T-Norm verifier [4] is used to verify target and impostor speakers. This is done by computing log-likelihood scores as follows:

$$score = (TS - mean(\mathbf{IS}))/stdev(\mathbf{IS}) \tag{2.6}$$

where TS is a log-likelihood score obtained by fitting the test sequence to the target model, **IS** is a vector of log-likelihood scores obtained by fitting the test sequence to each of the impostor models, $mean(\mathbf{IS})$ is the average of **IS** and $stdev(\mathbf{IS})$ is the standard deviation of **IS**. The test sequence is classified as a target sequence if the log-likelihood score exceeds a chosen threshold. The threshold is chosen according to the specific verification problem. If the task is of such a nature that it is more important to falsely reject target speakers than to falsely accept impostor speakers, the threshold would be chosen relatively high. A target speaker is falsely rejected if it is classified incorrectly as an impostor speaker. In the same way, an impostor speaker is falsely accepted if it is classified incorrectly as a target speaker. In another classification task, it might be more important to falsely accept impostor speakers than to falsely reject target speakers. In such a given task, the threshold would be chosen relatively low. The overall accuracy is computed by adding the number of incorrect classifications and dividing them by the total number of trials.

2.3.2 Detection Error Trade-off (DET) Curves

The T-Norm verifier generates scores, verifying the test trial against the target model and impostor models. The test trial can either be a target trial (generated from the target speaker) or an impostor trial (generated from a non-target speaker). The T-Norm scores can then be classified by setting up a threshold, where scores that are greater than the threshold are classified as "true", and scores that are smaller or equal to the threshold are classified as "false". With the pre-knowledge of which trials are actual target trials and which are actual impostor trials, we subsequently have four categories of classification:

- A target trial classified correctly as "true".
- A target trial classified incorrectly as "false", referred to as a false rejection (FR).
- An impostor trial classified correctly as "false".
- An impostor trial classified incorrectly as "true", referred to as a false acceptance or false alarm (FA).

Two types of errors can occur. The false rejection rate (FRR) is the percentage of target trials that are classified incorrectly, i.e. the percentage of "false" classifications of target trials, $FRR = P(FR|Target)$. The false alarm rate (FAR) is the percentage of impostor trials that are classified incorrectly, i.e. the percentage of "true" classifications of impostor trials, $FAR = P(FA|Impostor)$.

By sweeping through the likelihood scores and using different thresholds, it is possible to determine FAR and FRR at different operating points. The detection error trade-off (DET) curve [34] is a plot of these two types of error rates, FAR and FRR, on the x and y axes using a normal deviate scale. DET curves have the property that if the underlying distribution of scores for both target and impostor trials are Gaussian, the resulting performance curve is a straight line. The point on the DET curve where FAR=FRR is referred to as the equal error rate (EER).

For more information on the effect of the T-Norm type of normalisation on the DET curve see [39].

2.4 Literature Study

2.4.1 Acoustic

Speaker recognition is divided into two specific tasks, depending on the application: speaker verification and speaker identification. Either of these tasks can be divided into text-dependent (constrained to a known phrase) or text-independent (totally unconstrained) speaker recognition tasks. Speaker identification can either be closed-set (identification is restricted to a known group of speakers) or open-set (no restrictions - can be identified as "not part of the group of known speakers"). In speaker recognition tasks thus far, several types of databases have been used, i.e. *clean* speech databases (low noise level), such as the TIMIT database, or telephone speech databases (with a much higher noise level), such as NTIMIT database. It is important to note that noise degrades the performance of speaker recognition. Other types of speech, such as conversational speech, are provided in Switchboard.

Features

Up till recently, speaker recognition has commonly been done using an acoustic feature set, such as spectrum-based features and pitch. These features represent the physical aspects involved in speech, such as the vocal tract shape. The more popular feature extraction approaches use cepstral features [3].

In [55], the bispectrum, which is a higher-order statistical feature, is used for more robust speaker identification in various noise conditions. Different noise cases were examined by contaminating the training and testing data with the same type of noise: 10 dB additive white Gaussian noise and 10 dB additive coloured Gaussian noise. Using 20 speakers of TIMIT and a windowing frame length of 32 ms, the results obtained when using the bispectrum feature were 82.50% for white Gaussian and 80.5% for coloured Gaussian noise. This is quite an improvement over the result when using the cepstrum feature: 65.75% for both coloured and white Gaussian noise. However, the bispectrum did not perform as well when NTIMIT data was used, possibly because phase relations were distorted via the communication systems and formants below 300 Hz were removed.

In [50], statistics of pitch are used for prosody-based speaker recognition. In prosody-based speaker recognition, the types of utterance such as questions and statements, and people's attitudes and feelings are studied. The elements of prosody are derived from the

acoustic characteristics of speech, such as the pitch or frequency, the length or duration, and the loudness or intensity of speech.

Approaches

One of the first approaches for speaker recognition was to use long-term averages of acoustic features, such as spectrum reflection coefficients and pitch to average out factors influencing acoustic features such as phonetic variations. This leaves only the speaker-dependent component [33].

Another approach to speaker recognition is to explicitly model the speaker-dependent acoustic features within the individual phonetic sounds. The speech is segmented into phonetic sound classes prior to speaker model training. This approach is attractive, because different phonemes have different levels of usefulness for speaker recognition. In [41, 48, 27], the speech is segmented into phonetic categories, while in [14], the speaker recognition system is based on vowel-spotting. A segmental approach to speaker recognition can also be used to discard or de-emphasise parts of speech that are contaminated with background noise, channel artifacts and cross-talk [19]. This contamination degrades the performance of speaker identification systems. In [57], a database of speakers engaged in dialogue (Switchboard) is used. Segments of speech from the same speaker are automatically grouped together (clustering) and used for speaker identification. Clustering performance improves with the length of the segments being clustered. The performance increases from 80% correct (using segments of 0.4 to 0.8 seconds duration) to over 90% correct (using segments of over 2 seconds duration).

The *Gaussian Mixture Model* (GMM) used for speaker recognition falls into the implicit segmentation approach to speaker recognition. It is a probabilistic model that models the underlying speech sounds of a speaker's voice. GMMs have been used for both speaker verification and identification systems [45, 30, 8]. In [52], a combination of GMM output probabilities is used to generate decision rules.

The *Hidden Markov Model* (HMM) is another probabilistic model that, like the GMM, also models the underlying speech sounds, but differs from the GMM in that it also models the sequencing among these sounds. HMMs (typically left-to-right) are commonly used in text-dependent speaker recognition tasks in various configurations or in combinations with other models [46, 7, 37, 54]. For text-independent tasks, the sequencing of sounds in the test data is not necessarily reflected in the training data. In [35], ergodic mixture Gaussian HMMs are used for text-independent speaker recognition. Their experimental results

show that the information on transitions between different states is not effective for text-independent speaker recognition. This conclusion was arrived at because the identification accuracy was dominantly dependent on the total number of mixtures (number of states times the number of mixtures).

The most recent approach to speaker recognition is the use of *neural networks* (NN). It differs from the GMM and HMM approaches in that it does not train individual models to represent speakers, but discriminative NN's are trained to model the decision function which best discriminates speakers within a known set. There are several types, such as the *modified neural tree network* (NTN) [15], *time-delay NN's* (TDNN) [5], *radial basis function networks* [40] and Predictive Neural Networks (PNN) [22]. In [47], a binary partitioned approach using NN's is used which improves the training times of the NN.

In [21], a *Nearest-Neighbour Distance Measure* (NNDM) is being used. The NNDM method is so termed because it is based on the measured distances from each frame of an utterance to the nearest other frame of the same utterance and to the nearest frame of every other utterance being compared.

Table 2.1 contains a summary of the accuracies for speaker recognition obtained, using different speech databases and approaches over different periods of time. Note that these results should not be directly compared to one another, since the databases used to obtain them differ in speech quality and quantity and should be taken into account. Results obtained from speaker identification and speaker verification cannot be directly compared, owing to the difference in recognition tasks. The results are given chiefly to form an idea of the research in the speaker recognition field over the past few years.

Using neural networks for speaker recognition on a high-quality database such as TIMIT, one can expect results of as high as 100%. In Table 2.1 such typical results are shown (of [47] and [22]). Mel-frequency cepstrum coefficients (MFCC) are far more popular features for speaker recognition than ordinary cepstral coefficients or linear predictive cepstral coefficients (LPCC) [10]. Another pair of interesting results in Table 2.1 is the last two entries of [14]. These results illustrate the huge effect that noise has on speaker recognition in the drop of the result of 98.09% in high-quality speech (TIMIT) to that of 59.32% in noisy speech conditions (NTIMIT).

Citation	Year	Database	Type	Features	Model	Accuracy
[46]	1991	20 speakers of telephone speech database	TDV	LPCC	HMM	96.5%
[5]	1991	20 speakers from TIMIT	TII	16-th order LPC coefficients	TDNN	98%
[47]	1991	47 speakers from TIMIT	TII	cepstral coefficients	NN's	100%
[22]	1992	24 speakers from TIMIT	TII	MRR[a]	PNN	100%
[54]	1993	microphone speech of 963 speakers	TDI	LPCC	HMMs	97.8%
[21]	1993	24 speakers from Switchboard	TII	MFCC	NNDM	95.9%
[21]	1993	51 speakers from KING	TII	MFCC	NNDM	79.9%
[37]	1994	100 speakers of telephone speech database	TDV	LPCC	HMM-MLP	93.6%
[45]	1995	49 speakers from KING	TII	MFCC	GMMs	96.8%
[30]	1997	88 speakers from Switchboard	TIV	MFCC	GMMs	84.3%
[52]	1997	45 speakers from Spidre	TII	MFCC	GMMs	91.1%
[14]	1997	410 speakers from TIMIT	TII	MFCC	GMMs	98.09%
[14]	1997	410 speakers from NTIMIT	TII	MFCC	Gaussian Mixture HMM	59.32%

Table 2.1: *Some results of speaker recognition between 1991 and 1997, showing the databases and models used. Recognition task types are indicated: TII — Text-independent speaker identification, TDI — Text-dependent speaker identification, TIV — Text-independent speaker verification, TDV — Text-dependent speaker verification.*

[a]mean rate response of the auditory model proposed by Sennef using 40 channels

2.4.2 Non-Acoustic

A very recent approach to speaker recognition is to use non-acoustic features such as phoneme or word labels. This approach differs from the acoustic approach mainly in that it models the usage of phoneme strings or words of the speakers rather than the differences of voice quality.

Phonetic Speaker Recognition

Andrews et al. does language-independent speaker recognition using phonetic features in [1]. Phonetic information from six languages is used to perform text-independent speaker recognition.

The *National Institute of Standards and Technology* (NIST) has included an Extended Data Speaker Recognition Evaluation Task that contains a large amount of training data. The purpose of this task is to foster new research on improving speaker recognition performance by investigating higher-level (non-acoustic) characteristics of speech. For the task in [1], the Switchboard I corpus provided by NIST during their 2001 task is used. The training data is split up into one, two, four, eight and sixteen conversation sides. A single channel conversation side contains speech from one of the two people taking part in a conversation and has a nominal length of 2.5 minutes. NIST makes use of a jackknifing procedure to cycle through the training and testing conversations to ensure an adequate number of tests. Switchboard I consists of a total of 483 unique speakers and 58 642 test conversations.

Phonetic speaker recognition is performed in four steps. First, a phoneme recogniser processes the test speech utterance (in the appropriate language) to produce classified phoneme label sequences. These phoneme sequences are then converted to N-gram frequency counts. The test N-gram counts are compared to a hypothesised speaker model and the Universal Background Phoneme Model (UBPM). Finally, the scores from the hypothesised speaker models and the UBPM are combined to form a single recognition score.

The algorithm used for the phoneme recogniser calculates twelve cepstral and thirteen delta-cepstral features on 20 ms frames with 10 ms updates. The cepstra and delta-cepstra are modelled using HMMs, and the HMMs are trained on phonetically marked speech in six languages (using the OGI multi-language corpus). The output probability densities

for each observation sequence (cepstra and delta-cepstra) in each state are modelled as a mixture of six Gaussian densities.

In [2] Andrews et al. improves the above system by incorporating gender-dependent phone models and the pre-processing of the speech files to remove cross-talk. More sophisticated fusion techniques were developed for the multi-language likelihood scores, and the improved system reduced the equal error rate to less than 3.58%.

Navrátil et al. uses maximum-likelihood binary-decision tree models to do phonetic speaker recognition [38]. Phonetic speaker modelling using N-grams has its disadvantages: In order to capture a reasonably long time window, the model order (referring to N-grams) needs to be chosen correspondingly high. This results in exponential growth in the number of parameters of the model. This problem can be solved by:

1. Providing sufficiently large amounts of training data for each speaker.
2. Decreasing the model order.
3. Using smoothing techniques.

The binary-tree structure Navraátil et al. introduces, allows exploiting dependencies from longer contexts than that of typical N-grams, while keeping the number of free parameters under control. A recursive smoothing technique and an adaptation step in creating the tree models are used to deal with limited training data. The NIST 2001 Switchboard corpus is used for the experimental setup.

Jin et al. uses two speaker identification systems: one multilingual system and one single language multiple-engine system [26]. Text-independent speaker identification experiments are conducted on a distant-microphone database (30 speakers). The multilingual system uses phonetic sequences from phone recognisers trained on multiple languages (8 in total), making it somewhat language-independent. The multi-engine system uses 3 different English phone recognisers which are trained on speech recorded in vastly different conditions, namely: Switchboard (telephone, highly conversational), Broadcast News (various channel conditions, planned speech) and Verbmobil English (high quality, spontaneous). A perplexity score is used to match decoded phonetic sequences to each speaker's phonetic language model. The best N-gram performance is gained by using trigrams and has an equal error rate of approximately 5%. The use of trigrams outperforms the use of bigrams using 8 and 16 conversation sides. Experiments with bigrams, using less than 4

conversation sides outperform those done with trigrams. Using smoothing and adaptation, a relative reduction in EER, ranging between 10-60% (compared to the best N-gram system) is achieved across the different training conditions.

Jin et al. found that for the given choice of multi-engine system, the multiple English phone recognisers provide less useful information for the classification task than multiple language phone recognisers. The best identification, using 60-second test data, results in an accuracy of 96.7 %, integrating all 8 languages.

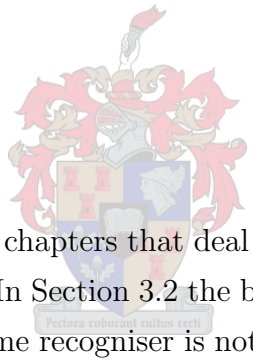
Idiolectal Speaker Recognition

In [11], word unigrams and bigrams are used to explore "familiar" speaker information. The NIST Switchboard I corpus is used to do speaker detection. The features are manual word transcriptions conducted by the *Institute for Signal Processing* (ISIP). These transcriptions are further processed to ignore punctuation and transcriber comments and to add start and end turn tags.

The model is trained on 1, 2, 4, 8 and 16 conversation sides. Each test uses a whole conversation side as the test segment. A log-likelihood ratio of true speaker likelihood to background speaker likelihood is used to test the data. An experiment performed to progressively prune the low-frequency bigram counts has found that a minimum threshold of 200 gives the best performance. The best result has an equal error rate of approximately 6.5 % and is obtained with 16 conversation sides (using a minimum threshold of 200).

Chapter 3

Incorporating Recogniser errors in the Modelling of Speaker Phonotactics



3.1 Introduction

This chapter is the first of three chapters that deal with the use of phoneme labels as feature set in speaker recognition. In Section 3.2 the basic set-up for the phoneme recogniser system is discussed. The phoneme recogniser is not a perfect system and errors do occur. Section 3.4.3 deals with the various errors a phoneme recogniser can make.

We are interested in modelling the time-dependencies among the phoneme labels. Because of this, HMMs are chosen as speaker models. Section 3.5 deals with the model structure chosen for speaker modelling. Section 3.6 deals with experiments conducted to illustrate how these errors can be utilised to improve our speaker recognition system. First-order HMM experiments are compared to second-order HMM experiments. From the second-order experiments, one learns that in order to improve higher-order results one needs to deal with the problem of data scarcity.

3.2 Our Approach

The phoneme-recogniser system consists of digital signal processing (DSP) of the raw speech signal and the calculation of *Mel Frequency Cepstral Coefficients* (MFCC). We are

interested in time-dependencies of the phonemes, so making use of an HMM to model these time-dependencies. One way would be to directly use the MFCC as feature vector for the HMM (acoustic approach). Another way, used in this study, would be to first make use of a phoneme recogniser which converts the MFCC to phoneme labels (non-acoustic approach). These phoneme labels would then be used as feature vectors for the HMM. The HMM output PDF would typically be discrete in this case. The latter approach is much faster than using the MFCC directly. (The typical length of a phoneme is around 80 ms. If we were to calculate a 32 dimensional MFCC feature vector every 10 ms, and if we were to use diagonal covariance for the GMM of the HMM PDF, it would take approximately 250 times slower using the MFCC directly as input to the speaker HMM than to use classified phoneme labels as input to the speaker HMM.) Figure 3.1 illustrates the system described above. The dashed arrow shows that the MFCC could be used directly as feature vector to the HMM. The preferred input to the HMM is the phoneme labels, illustrated with the black arrow.

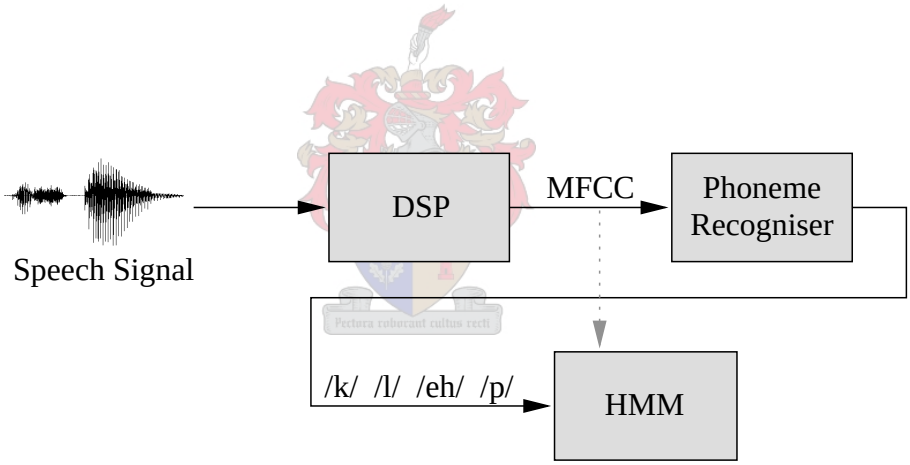


Figure 3.1: *The phoneme recogniser system, illustrating processing of the speech signal and calculation of MFCC that in turn are the input to the phoneme recogniser. The end result is phoneme labels that are used as the feature vector to an HMM.*

3.3 Databases

3.3.1 TIMIT Database

The TIMIT database was developed using many speakers and continuous speech, The selection of speech was carefully controlled and contained dialects around the Continental

United States. It has a carefully designed breadth and depth of phonetic coverage. The database contains 630 speakers each uttering 10 sentences, making a total of 6300 sentences. The recorded utterances were orthographically and phonetically transcribed. The speech was recorded in careful acoustically controlled conditions with a high quality wideband microphone, making it undesirable for telephone bandwidth speech analysis [24].

3.3.2 NTIMIT Database

The NTIMIT database was generated by transmitting the TIMIT database over a long-distance telephone network. It is therefore orthographically and phonetically equivalent to the TIMIT database. It was transmitted in an acoustically isolated room through an "artificial mouth" designed to approximate the acoustic characteristics of the human mouth and using a device to approximate acoustic coupling between the human mouth and the telephone handset.

Transmission of utterances to various locations was achieved by using a "loopback" telephone path to a large number of central offices, varying the geographic location of the central office that the utterance was transmitted to. Half of the database was transmitted over local telephone paths, while half was transmitted over long-distance paths [24, 36].

3.3.3 1996 ICSI Transcriptions

In the *Switchboard* corpus, two individuals discuss a specific topic for several minutes over the telephone. The *International Computer Science Institute* (ICSI) used a subset of 72 minutes of the Switchboard corpus (618 conversations from 750 speakers) for phonetic transcription. The subset was phonetically transcribed by students with a background in phonetic transcription. These students were supervised to ensure as accurate and as uniform transcriptions of the materials as possible. The phonetic transcriptions were encoded with a variant of the Arpabet transcription system used for the TIMIT corpus. The transcription was augmented with a set of diacritics representing such phonetic properties such as glottalisation ("creaky voice"), nasalisation, frication, aspiration, de-voicing, unusual voicing, and velarisation. In addition, transitional elements between adjacent vocalic or glide-like segments were explicitly marked. In total there were 56 phones used to transcribe the Switchboard corpus [20].

3.3.4 Switchboard I and II Corpus

The *Linguistic Data Consortium* is an open consortium of universities, companies and government research laboratories, which creates, collects and distributes speech and text databases and other resources for research and development purposes[1]. The Switchboard I Telephone Speech Corpus was originally collected by Texas Instruments in 1990 to 1991, under DARPA sponsorship. The first release of the corpus was distributed by the LDC in 1992 to 1993.

The Switchboard I corpus is a collection of about 2400 two-sided telephone (landline) conversations among 543 speakers (302 male and 241 female) from all areas of the United States. All conversations of Switchboard are in English and the speech is sampled at 8 kHz. The Switchboard calls were handled automatically, giving the caller appropriate prompts in which another person is selected and dialled to take part in a conversation. A topic was introduced for discussion and the speech of the two speakers was recorded on two separate channels until the conversation ended. Approximately 70 topics were provided of which roughly 50 were used frequently. The selection of the topics was constrained so that

1. no two speakers would take part in a conversation more than once and
2. no one spoke more than once on a given topic

During the collection of speech (by the LDC) for Switchboard II, each speaker was asked to participate in at least 10, five minute telephone (landline) calls. A topic for discussion was given, although participants had the freedom to discuss anything they wished. The participants placed their calls via a toll-free robot operator. Each participant had a personal identification number (PIN) with which they obtained access to the robot operator. Particular attention was paid to PIN verification (matching speaker with PIN) by the LDC staff.

Switchboard II Phase I consisted of 3638, five minute conversations involving 657 speakers (299 male and 358 female). Potential speakers responded from all over the United States, although the majority were from the Mid-Atlantic area. Participants in Swithboard II Phase II were recruited from mid-western college campuses. There was a total of 679 participants. The Switchboard II Phase III collection was focussed primarily in the American

[1]The following information can be found on *http://wave.ldc.upenn.edu*

South. The project's goal was to target native speakers of English in the American South. The LDC collected a total of 2728 calls from 640 participants (292 male and 348 female).

3.4 Phoneme Modelling

3.4.1 Feature Extraction

We calculate 12-dimensional MFCC features from the raw speech signal[2]. Cepstral mean subtraction is used to improve robustness to adverse channel effects. The feature vector is augmented with velocity (delta) and acceleration (delta-delta) features, increasing the dimension to 36. After dimension reduction with *linear discriminant analysis* (LDA), the final feature vector has 19 dimensions. LDA is a method that reduces the dimensions of the feature vector, while it maximises class separation [6].

3.4.2 Model Structure

Model: The phoneme recogniser consists of a phoneme spotter HMM containing several submodels.

Submodel: The submodel is a 3-state left-to-right HMM, each state having a GMM as output PDF.

PDF: The output PDFs associated with each state of the submodel are full covariance GMMs with eight components each.

3.4.3 Substitution, Insertion, Deletion (SID) Counts

The combined NIST Switchboard I ICSI and NTIMIT data set contains a set of genuine phoneme transcriptions (genuine labels) and a set of test transcriptions (classified labels) that are generated by the phoneme recogniser. This data set is used to compute substitution, insertion and deletion (SID) counts of the phoneme classifier.

[2]Andre du Toit from the DSP lab at the University of Stellenbosch designed a phoneme recogniser during the NIST 2002 evaluation task.

Consider the following genuine phoneme utterance:
ih *d* *aw* *ah*

If the classifier classifies the following utterance:
ih *k* *d* *aw* *ah*
'*k*' would be an **insertion**,

If it classifies the following:
ih *aw* *ah*
'*d*' would be a **deletion**,

and if it classifies
ih *d* *ay* *ah*
'*ay*' would be a **substitution**.

Let $GL(i)$ be the i-th genuine label in the set of genuine labels and $CL(j)$ be the j-th classified label in the set of classified labels. A substitution count is the number of times a genuine label is substituted with a classified label. A matrix of substitution counts $\mathbf{C} = c_{ij}$ can be generated so that index i in c_{ij} is associated with the genuine label $GL(i)$ and index j is associated with classified label $CL(j)$. A genuine and classified label that are the same are mapped to or associated with the same index. This is done to simplify the interpretation of c_{ij} so that when $i = j$ and i and j map to the same label, the substitution count c_{ij} indicates correct classification of the phoneme recogniser. For instance, the substitution count where genuine label /a/ is substituted with classified label /a/ indicates correct classification.

The substitution error probabilities (SEP) of genuine labels with classified labels can be calculated by normalising $\mathbf{C_{ij}}$ over i:

$$P(CL(j)|GL(i)) = c_{ij} / \sum_{i=1}^{M} c_{ij} \qquad \forall i \tag{3.1}$$

where M is the number of genuine labels. $P(CL|GL(i))$ is the SEP of genuine label $GL(i)$. By doing this, a confusion matrix is calculated. The probabilities of correct classification are indicated where $i = j$ and i and j map to the same label. All other probabilities are substitution error probabilities.

The substitution error probabilities are used to model the substitution errors of the phoneme recogniser in the PDF of the HMM and is explained in Section 3.5.4.

3.4.4 Incorporating the Databases with the Phoneme Modelling

The phoneme spotter HMM contains 39 phonemes derived from the TIMIT phoneme set. The phoneme submodels are 3-state left-to-right phoneme HMMs and are trained, using the NIST Switchboard I ICSI transcriptions combined with NTIMIT. The phoneme transcriptions are mapped to the 39 phonemes used in the SPHINX system [31]. The average phoneme transcription accuracy is 54% on the combined NIST Switchboard I and NTIMIT sets. The accuracy is calculated as 100% - the phoneme error rate [23]:

$$\text{Phoneme Error Rate} = \frac{Subs + Dels + Ins}{\text{Number of phonemes in speech segment}} \tag{3.2}$$

where $Subs$ is the number of substitutions, $Dels$ is the number of deletions and Ins is the number of insertions (see Section 3.4.3).

3.5 Speaker Modelling

3.5.1 Universal Background Model

Since data scarcity is a problem in this study, it is appropriate to train a *Universal Background Model* (UBM) to assist in the training of target (hypothesised) speakers by acting as initial model or prior model for the target speakers. The UBM speakers are selected from a set of speakers that is not included in the set of target speakers. To train the UBM, all the speakers of the NIST Switchboard II corpus that are excluded from the set of target speakers of jackknife set 0 (jack_0) are used. (This set is the same as the set of target speakers from jackknife set 1 (jack_1) to jackknife set 9 (jack_9)). The UBM is trained using the pooled data from the set of UBM speakers. The same model type and model structure are chosen for the target speakers, impostor (non-target) speakers, and UBM speakers.

3.5.2 Training and Evaluation Setup

As part of its 2002 evaluation task, the *National Institute for Standards and Technology* (NIST) has provided the following information in its evaluation plan for 2002, available

on its website[3]: During 2002, NIST used the Switchboard II[4] version 2 (Phase II and III) corpus that serves as the primary data set for the extended data evaluation. The extended data evaluation consisted of longer speech patterns in order to research higher-level speech information to improve speaker verification performance. NIST also provided a speaker-conversation table and an evaluation control file to define the training and the evaluation test sets. The speaker-conversation table is a file that gives the conversation sides for each speaker in the corpus. A conversation side is one side of a conversation of a specific speaker taking part in a conversation. A conversation side in Switchboard II has a nominal length of 2.5 minutes. A conversation-identifier specifies a set of conversation sides containing speech from a specific speaker. The conversation-identifiers consist of speakers with 4, 8, and 16 conversation sides.

The evaluation control file supervises the evaluation. It controls the training of models and defines the testing thereof. It makes use of a jackknifing technique that rotates training and test data in order to provide an adequate number of tests. The control file is structured in such a manner that it can accommodate systems that create a background model to assist the training of the target model. Since the target and test data are rotated in the jackknifing scheme, it is necessary to control the training of the background model, the target models and the test trials to ensure unbiased testing. Each jackknife block (10 altogether) has an evaluation control file. The speaker-control file provides several conversation sides from a speaker so that there are multiple models for each speaker. The test-sides contain test segments from both the target speaker and impostor speakers.

The models are set up in such a way that speakers with models trained from 8 conversation sides also have models trained from 4 conversation sides. Speakers with models trained from 16 conversation sides also have models trained from 4 and 8 conversation sides. Table 3.1 shows the data description of Switchboard II, including the data of all jackknife blocks.

For the experiments in this chapter and the ones to follow, we make use of the data corpus of Switchboard II (version 2), and adhere to the evaluation control files provided for the NIST 2002 evaluation task. The experiments in this chapter are done using the first jackknife block (jack_0).

---

[3]See *http://www.nist.gov/speech*

[4]See Section 3.3.4.

Number of Training Conversation Sides	Number of Target Models	Number of Unique Speakers	Number of Test Trials
4	1542	671	26246
8	1551	604	23009
16	1420	230	10710

Table 3.1: *NIST extended data description for Switchboard II, version 2*

3.5.3 Model Structure

As we wish to model time-dependencies of the phoneme labels, we choose an HMM (described in Section 2.2.2) to model the speakers. The same basic model structure is used for the UBM, target and impostor speakers. Experiments are conducted, using various means of model initialisation and training. The classified phoneme labels at the output of the phoneme classifier are used as feature vectors for the HMM. The first-order HMM has the following structure:

- An HMM with a fully connected ergodic structure is used. This particular structure is described at the end of Section 2.2.2.
- The HMM is discrete, with one state per phoneme label.
- Different types of initialisation for the PDFs and transition probabilities are used (See Section 3.5.4). During training we train only the transition probabilities of links that connect emitting states to emitting states. The transition probabilities from the initial state or to the terminating state are not trained. The PDFs are initialised using the SEP of the phoneme recogniser and stay fixed during training.

The model structure of the second-order HMMs is similar to that of the first-order HMM. First-order equivalents of second-order HMMs can be computed by making use of the ORED algorithm [12]. These equivalent models have more than one state associated per phoneme label. Each state associated with a particular phoneme label shares the same PDF. The transition probability matrix of a second-order HMM can be described by $\mathbf{A} = [a_{hij}]$, and that of a first-order by $\mathbf{A} = [a_{ij}]$ (see Section 2.2.2). The second-order HMM is initialised by setting $a_{hij} = a_{ij}$ for all h.

3.5.4 Initialisation and Training

There are different ways to initialise the transition probabilities of the UBM:

1. One way would be to use equal probabilities for all the transition probabilities leaving a specific state. The short notation for this type of initialisation (modelling) is referred to as EP in this study.

2. A bigram count of phonemes is the number of times a sequence of two specific phonemes occurs. Another way would be to initialise the transition probabilities with normalised bigram counts associated with a given state[5]. The bigrams are normalised in much the same way as SEP in Equation 3.4.3. The short notation for this type of initialisation (modelling) is referred to as BG in this study.

One can also initialise the PDF of the UBM in, amongst others, the following two ways:

1. One can model SEP of the phoneme recogniser by using Equation 3.4.3 and letting the PDF $= P(CL|GL(i))$. This is a way to define the erroneous outputs of the phoneme recogniser and employing them in the HMM. The HMM states are linked to the genuine phoneme labels and the PDF to the classified phoneme labels. A classified output label of the phoneme recogniser is therefore not necessarily the correct one. Knowing this, the modelling of SEP so to speak "opens" up other possible Viterbi output paths that could give better speaker recognition results.

2. If one does not model the SEP[6], the PDF is initialised by a unity matrix, so that the PDF $f(x|GL(i))$ of a given state[7]

$$f(x|GL(i)) = \left\{ \begin{array}{lll} 1 & i = j, & \text{Labels GL(i) and CL(j) are common} \\ 0 & i \neq j, & \text{Labels GL(i) and CL(j) differ} \end{array} \right\} \tag{3.3}$$

The UBM is typically used to initialise the first-order target models. (Correspondingly, one can also initialise second-order target models with a second-order UBM.) Figure 3.2 shows the basic procedure followed in this chapter for the initialisation and training of target and impostor speakers. The initial set of parameters for the UBM, λ_0, is at the

[5]These bigram counts are calculated using data from speakers included in the UBM.

[6]In this case the state sequence in the HMM is not hidden, and is reduced to a Markov Chain.

[7]The states are associated with genuine phoneme labels.

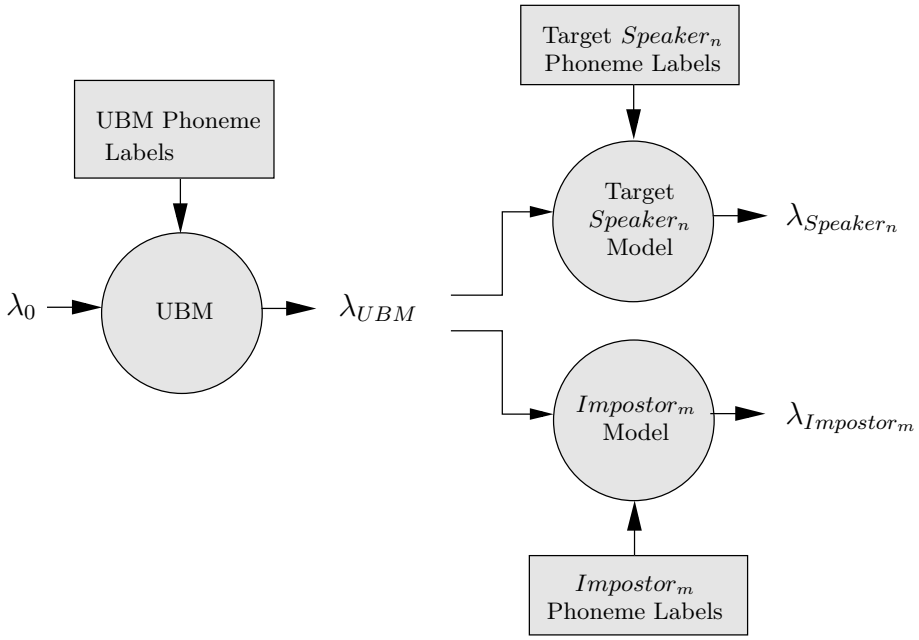


Figure 3.2: *Basic representation of the initialisation and training process followed for first-order HMM speakers and impostors.*

input of the UBM in Figure 3.2. These conditions depend on the method of initialisation of the PDF and the transition probabilities of the HMM (either EP or BG, as discussed in Section 3.5.4). The pooled data of the UBM speakers is trained to give the set of output parameters, λ_{UBM}. Both the speaker models and the impostor models are initialised in exactly the same way with the output parameters of the trained UBM. The target speaker models (TSMs) are trained for each target $speaker_n$ to give the output parameters $\lambda_{Speaker_n}$ for each speaker, where n indexes the target speaker in the target set. Each $impostor_m$ is trained to give the output parameters $\lambda_{Impostor_m}$, where m indexes the speaker in the impostor set.

3.5.5 Verification

The T-Norm verifier described in Section 2.3.1 is used for the verification of experiments in this study. The impostor models for the T-Norm verifier are trained from the data in NIST Switchboard I. A set of 40 speakers, each having 16 conversation sides, is chosen. During verification, the impostor models (trained using 16 conversation sides) are used in all the experiments (including those where target models are training using 4 and 8

conversation sides). This causes a slight mismatch between TSMs with a different number of conversation sides than those of the impostor models. Results would most likely improve when verification is done using impostor models with the same number of conversation sides as the TSMs. Choosing a set of impostors with 16 conversation sides is done merely for the sake of simplicity.

The first-order impostor models are used in both first-order and second-order experiments. (First-order experiments use first-order HMMs to model the target speakers, and second-order experiments use second-order HMMs to do so.) Using first-order impostor models for second-order experiments is much faster than using second-order imposter models. (Depending on the number of links in the HMM, this can be up to 45 times faster). The choice of impostor models may influence the results. For instance, a choice of first-order impostor models would most likely obtain different results than a choice of second-order impostor models in second-order experiments.

3.6 Experiments

All following experiments use the model structure discussed in Section 3.5.3, the T-Norm verification discussed in Section 2.3.1 and the verification process in Section 3.5.5. For initialisation of the transition probabilities of the UBM, we use either EP or BG as discussed in Section 3.5.4. We also try out different initialisation of the PDF by either modelling SEP or not (refer to Section 3.5.4). The experiments are done using the 265 target speakers in jackknife set 0 (jack_0).

3.6.1 Modelling of Substitutions vs no Modelling of Substitutions in the PDFs

In this experiment the two different types of initialisations of the PDFs of the UBM (discussed in Section 3.5.4) are used. The first one models SEP in the PDF, and the second one does not. Both of these HMMs are initialised with BG as explained in Section 3.5.4. The initial UBM models are then trained with the speakers from the background set.

Figure 3.3 shows the detection-error trade-off (DET) curves (see Section 2.3.2) for modelling of SEP vs no modelling thereof, for target models using 4 conversation sides for training. The EER (FAR=FRR) is where the diagonal dotted line on the DET plot intersects with the DET curve.

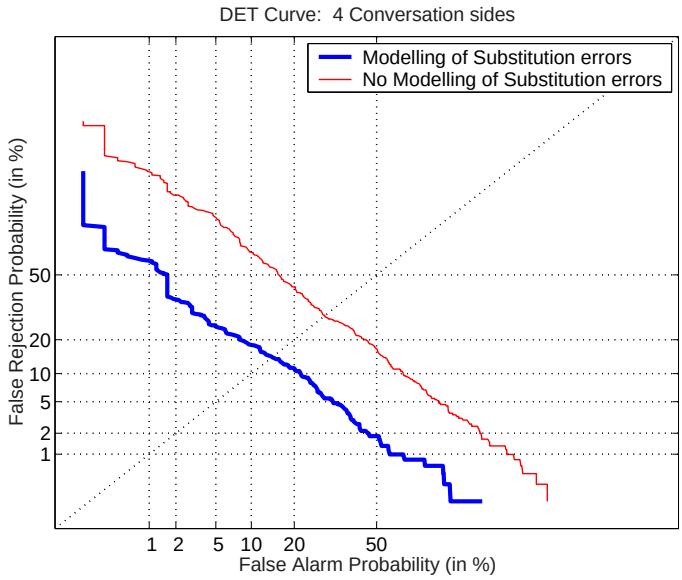


Figure 3.3: *DET curves for modelling of substitution errors and no modelling thereof, using 4 conversation sides for training.*

	EER		
	4 CS	**8 CS**	**16 CS**
Modelling of SEP	14.26%	11.23%	9.30%
No Modelling of SEP	29.89%	22.96%	12.27%

Table 3.2: *EERs of modelling of SEP vs no modelling of SEP, using target speakers trained on 4, 8 and 16 conversation sides (CS).*

Table 3.2 shows the EERs of no modelling of SEP vs modelling of SEP for training done with speakers with 4, 8 and 16 conversation sides (CS). The first thing to be noticed is that the results improve when more conversation sides are used for training. The EERs of the speakers that are trained using 4 conversation sides are more than halved when modelling SEP compared to when not modelling it. Modelling SEP is a great improvement of the phonetic speaker recognition system as was anticipated in Section 3.5.4. (Results when using 8 and 16 conversation sides for training also show an improvement when modelling SEP. These results are shown in Figures A.1 and A.2.)

Summary

- Results improve with an increasing number of conversation sides, since model estimation is better when more data is used for training.
- The phoneme recogniser has substitution, deletion and insertion errors. These errors contain useful information about the phoneme recogniser and can be modelled.
- Modelling of SEP gives a vast improvement in results compared to results where there is no modelling of SEP.

In all the preceding experiments, the substitution errors of the phoneme recogniser are modelled in the PDFs of the HMMs, based on the results obtained in this section. Initialisation of the HMM transition probabilities is done using both BG and EP, as described in Section 3.5.4.

3.6.2 First-Order vs Second-Order HMMs

We are interested in modelling even longer time-dependencies among the phoneme labels. This is done by setting up an experiment that compares the results of target models trained with first-order HMMs to those trained with second-order HMMs. For the first-order HMM experiments, the training method illustrated in Figure 3.2 is used. First-order impostor models are used in both first-order and second-order experiments as explained in Section 3.5.5.

Figure 3.4 shows the DET curves of first-order (X1(EP)) and second-order (X2(EP)) experiments with initialisation of transition probabilities using EP, using 16 conversation sides for training.

The second-order results show a decrease in accuracy compared to the first-order results. This is due to not having enough data for sufficient second-order model training. Data scarcity becomes more of a problem with higher-order models, as there are more parameters to estimate. The amount of data per parameter is less for higher-order models, hence the bigger problem with data scarcity. The same is found when using 4 and 8 conversation sides for training. (Figures A.3 and A.4) The results of first-order and second-order experiments using BG are much the same as those using EP. (Figures A.5, A.6 and A.7).

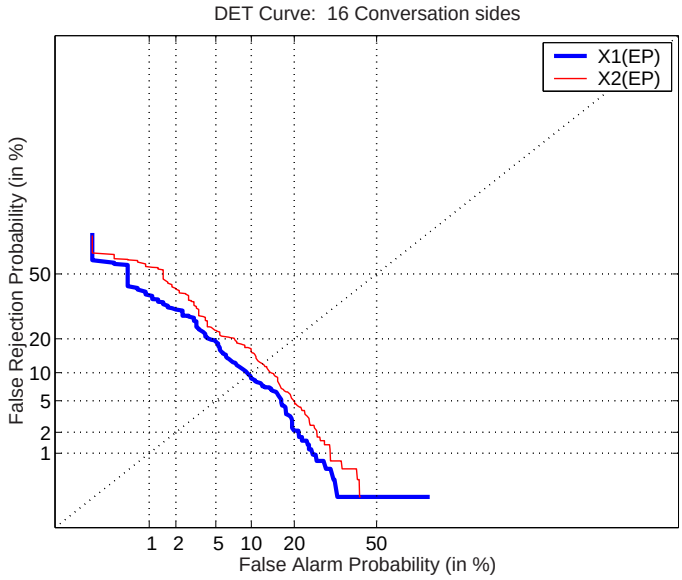


Figure 3.4: *DET curves for first-order, X1(EP), and second-order experiments, X2(EP), using 16 conversation sides for training: Initialisation with equal probabilities.*

Summary

Higher-order models have more parameters to estimate and therefore need more data to ensure sufficient model estimation. Data scarcity is a problem in these models and needs to be addressed.

3.6.3 Initialisation of Transition Probabilities

In this experiment the two initialisation methods, BG and EP, that are used to initialise transition probabilities are compared. These methods are described in Section 3.5.4. Initialisation of BG and EP are compared for both first-order (Figure 3.5) and second-order (Figure 3.6) results.

Conclusion

From these figures it can be seen that modelling is not sensitive to the two types of initialisations (BG or EP).

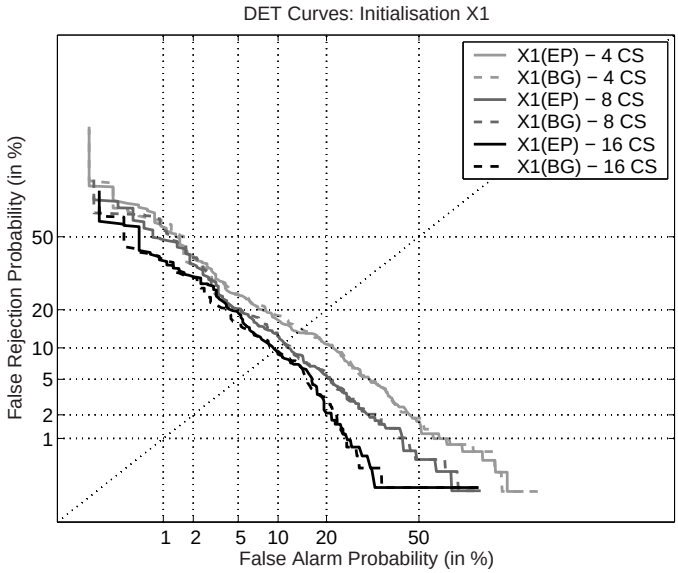


Figure 3.5: *DET curves comparing initialisation of EP and BG of first-order experiments.*

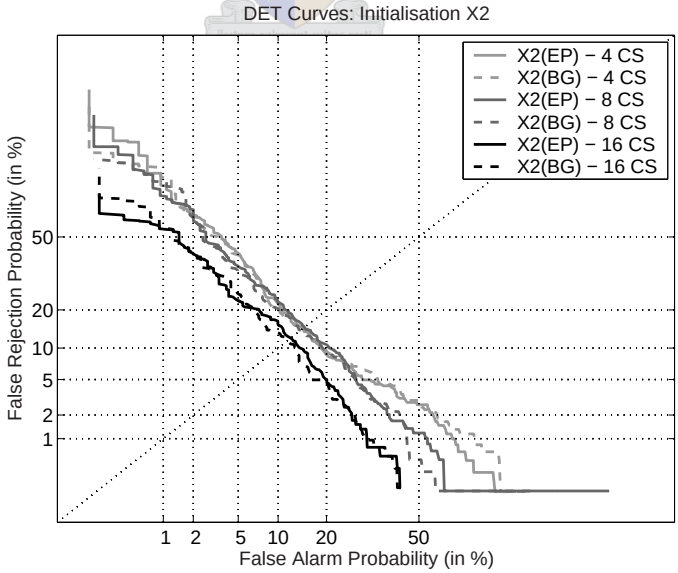


Figure 3.6: *DET curves comparing initialisation of EP and BG of second-order experiments.*

3.7 Summary

Modelling the SEP of the phoneme recogniser is a tremendous improvement of the phonetic speaker recognition system. Without the use of smoothing, the second-order results performs worse than the first-order results. The two initialisation methods (BG and EP) that initialise the transition probabilities produce very similar results.

In the next few chapters a number of solutions for the problem of data scarcity is investigated. Several means of smoothing techniques, such as decreasing the number of categories in the feature set, are tried, with the aim of increasing the amount of data for each category. By doing this, data is "gained", but specificity is lost. We investigate the smoothing of probability links of the HMM by using a Dirichlet estimator that uses well-trained models as prior models to smooth other models. A few possibilities for prior models are considered.

Chapter 4

Addressing the Problem of Data Scarcity

4.1 Introduction

In the previous chapter two speaker models are compared: One in which the PDF does no modelling of the SEP of the phoneme recogniser, and the other that takes these probabilities into account. Taking these substitution errors into account improved the speaker recognition system a great deal. For the experiments which follow, we therefore choose to model SEP in the PDF of the speaker HMMs.

In the previous chapter, when first-order speaker modelling is compared to second-order speaker modelling, data scarcity is seen to have a marked effect on especially the higher-order models. If one is to explore the use of higher-order speaker models, the problem of data scarcity needs to be addressed: This is what we do in this chapter. Two smoothing techniques are investigated: Dirichlet smoothing, and the use of fewer parameters by merging of the phoneme labels. The goal of the experiments is to establish which techniques are suitable for speaker recognition. As we want to form a basic idea of which of these techniques are promising, results are compared based on EERs.

4.1.1 Chapter Outlay

Section 4.2.1 deals with the merging of phoneme labels with the purpose of using fewer parameters in the speaker models. Section 4.2.2 describes the theory of the Dirichlet estimator used to smooth the transition probabilities of the HMM. In Section 4.3.3 the

UBM is used to smooth target models. In such a case we refer to the UBM as a prior model. First-order speaker models as prior models are also discussed.

The merging of phoneme labels can be used to train a model with fewer parameters. Such a model is referred to as a merged model and is trained to create more data per HMM link. This technique is described in Section 4.2.1. A merged model can also be used as prior model as discussed in Section A.5. The merging of labels and the use of merged models for smoothing ultimately proved to be of no significant value for speaker recognition. The material in Section 4.2.1 is included to provide a more complete picture of the work done. The use of prior models gives better results than merging labels. Section 4.3.3 is important, as it is the focus of Chapter 5.

4.2 Addressing the Data Scarcity Problem

The experiments of this chapter are based on the combination of two basic solutions to address the problem of data scarcity. The first solution is to use fewer parameters in the speaker models. The motivation for this and the approach taken are discussed in Section 4.2.1. The second solution is to smooth the transition probabilities of the HMM speaker model with those of another model (prior model).

This study makes use of Dirichlet estimation. In Section 4.2.2 a theoretical description of the Dirichlet estimator is given and in Section 4.2.2 the approaches taken in using prior models to smooth other models are discussed.

4.2.1 Using Fewer Parameters

One motivation for using fewer parameters in the HMM speaker models would be to optimise the training and evaluation speed. It is possible to determine which phoneme labels are more significant for speaker recognition than others, and use only the most important ones for the task of speaker recognition. Another possible way would be to merge some of the phoneme labels into one category. This not only helps increase the training and evaluation speed, but means that there is more data available per HMM link. This is not the case where some of the less significant phoneme labels are ignored for speaker recognition.

There are a few possible ways of merging phoneme labels. One would be to form broad categories based on continuant and non-continuant sounds. This entails the merging of all

the plosives into one category, all the affricates into another and so forth. Our approach is to merge the phoneme labels that cause the most confusion in classification. For instance, if a phoneme recogniser makes the most errors in confusing phoneme labels */ae/* and */eh/*, then these labels are merged into one category. The next few paragraphs explain how this is done.

Merging Labels

In any given classification problem, there are two sets of labels: The genuine label set which is the true labels, and the classified label set which is the labels at the output of the recogniser. The labels in these two sets are not necessarily the same.

The following definitions concerning these two sets are given:

- **Common Label (CL)** - a label that is common to both the genuine labels and the classified labels.
- **Classified-Only Label (COL)** - a label that appears only in the classified set.
- **Genuine-Only Label (GOL)** - a label that appears only in the genuine set.

When merging labels, one needs to decide which labels to merge, and the manner in which merging is to be done. We choose to merge labels that confuse the most, merging two labels at a time. This is done by calculating a matrix of SEP, $\mathbf{C} = c_{ij}$, as described in Section 3.4.3.

- If there are K common labels, the substitution counts associated with common labels is a $K \times K$ matrix.
- If there are M genuine labels, and N classified labels, then $\mathbf{C}$ is a $M \times N$ matrix.

Labels that cause the most confusion are labels with the highest values of c_{ij}, except at correct classifications of labels. Correct classifications of labels are indicated in c_{ij} where $(i = j <= K)$.

In the case where one wants to merge labels to decrease the number of links in the HMM, the genuine labels are merged with other genuine labels. (The states are associated with genuine labels). There are therefore two separate cases for merging labels when applied to decreasing the HMM links:

One can merge a **common label** with a **common label** and a **common label** with a **genuine-only label**. If the merging of labels is not directly applied to decreasing the HMM links, one can also merge a **common label** with a **classifed-only label** and a **classified-only label** with a **genuine-only label**. As regards to the set of phoneme labels in this study, the genuine phoneme set and the classified phoneme set are the same. This simplifies the merging to the merging of a common label with a common label. (All the other cases are discussed in Section A.3).

Figure 4.1 illustrates the substitution counts when merging a common label with a common label before and after label merging has taken place. The matrix of substitution counts before label merging is represented by c_{ij}. Genuine labels (rows) are indexed by i, while classified labels (columns) are indexed by j. Let c^*_{vw} represent the new substitution count, where genuine labels are indexed by v and classified labels by w. $L1$ and $L2$ are the labels that are to be merged. Genuine labels $L1$ and $L2$ are indexed by i_{L1} and i_{L2}, while classified labels $L1$ and $L2$ are indexed by j_{L1} and j_{L2}.

Vertical light grey areas indicate the substitution counts of a genuine label that is substituted with classified labels $L1$ or $L2$. Horizontal light grey areas indicate the substitution counts of genuine labels $L1$ or $L2$ that are substituted with a classified label . Dark grey areas indicate substitution counts where $L1$ is substituted with either $L1$ or $L2$, or $L2$ is substituted with either $L1$ or $L2$. Visually interpreted, these are the intersections of the vertical light grey regions with the horizontal light grey regions. Let L_{new} be the new merged label and L_{other} be any other label.

The calculation of the new substitution counts, when $L1$ and $L2$ are merged to a new label, L_{new}, takes place as follows (c_{ij} represents the matrix of substitution counts before label merging and c^*_{vw} represents it after label merging):

For the merging of a common label $L1$ with another common label $L2$, the new substitution counts are

$$c^*_{v_{L_{new}} w_{L_{new}}} = c_{i_{L1} j_{L1}} + c_{i_{L2} j_{L2}} + c_{i_{L1} j_{L2}} + c_{i_{L2} j_{L1}} \tag{4.1}$$

$$c^*_{v_{L_{other}} w_{L_{new}}} = c_{i_{L_{other}} j_{L1}} + c_{i_{L_{other}} j_{L2}} \tag{4.2}$$

$$c^*_{v_{L_{new}} w_{L_{other}}} = c_{i_{L1} j_{L_{other}}} + c_{i_{L2} j_{L_{other}}} \tag{4.3}$$

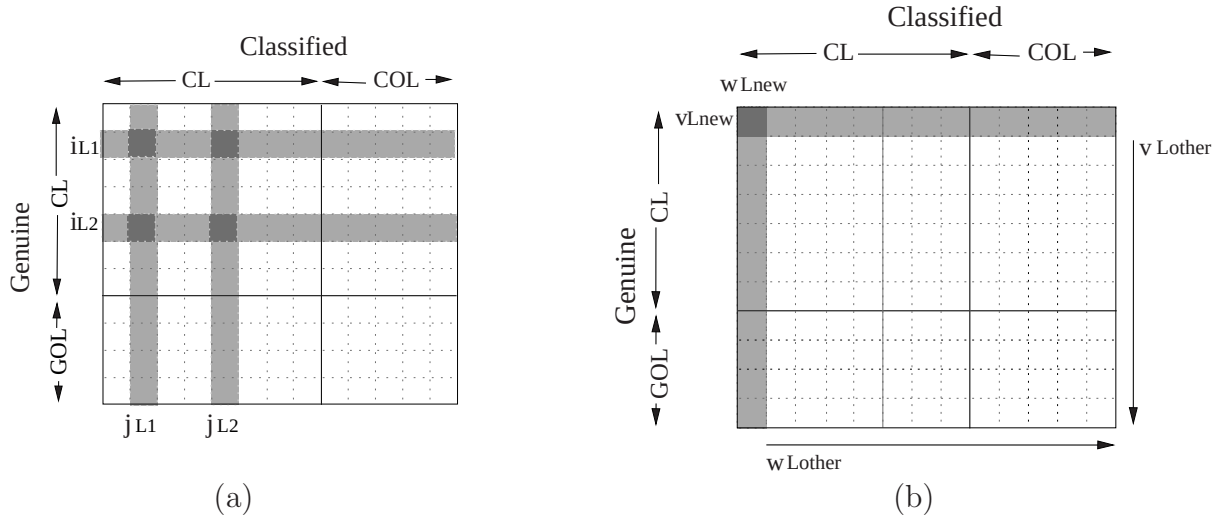


Figure 4.1: *A matrix representation of substitution counts before and after label merging of a common label with a common label takes place (a) Substitution counts c_{ij} before merging (b) Substitution counts c^*_{vw} after merging*

Computation of New Merged Feature Sets

Substitution counts are computed from the original feature set for the phoneme labels (39 phoneme labels), using a *dynamic time warping* (DTW) procedure [10]. From these counts we compute SEP, and merge two labels at a time, 5 times, using the procedure described in Equations 4.1 to 4.3 . When merging the labels, a new feature set is computed. Let us define this new feature set as the **merged** feature set, and the model computed from the **merged** feature set as the **merged** model. The new merged feature set is then used to compute a new set of substitution counts, c^*_{vw}, using a DTW procedure. This process is repeated to generate feature sets containing up to 9 unique phoneme labels.

Figure 4.2 illustrates a flowchart of the merging of labels and the generation of new merged feature sets. The process (marked with the letter A) shows the computation of substitution counts using a DTW procedure. The grey region (marked with the letter B) depicts the process of the merging of labels and the computation of new substitution counts described by Equations 4.1 to 4.3. One can decide how many times procedure B should be repeated before returning to procedure A. In the experiments to follow, procedure B was repeated five times, ie. merging two labels at a time, 5 times, before repeating procedure A.

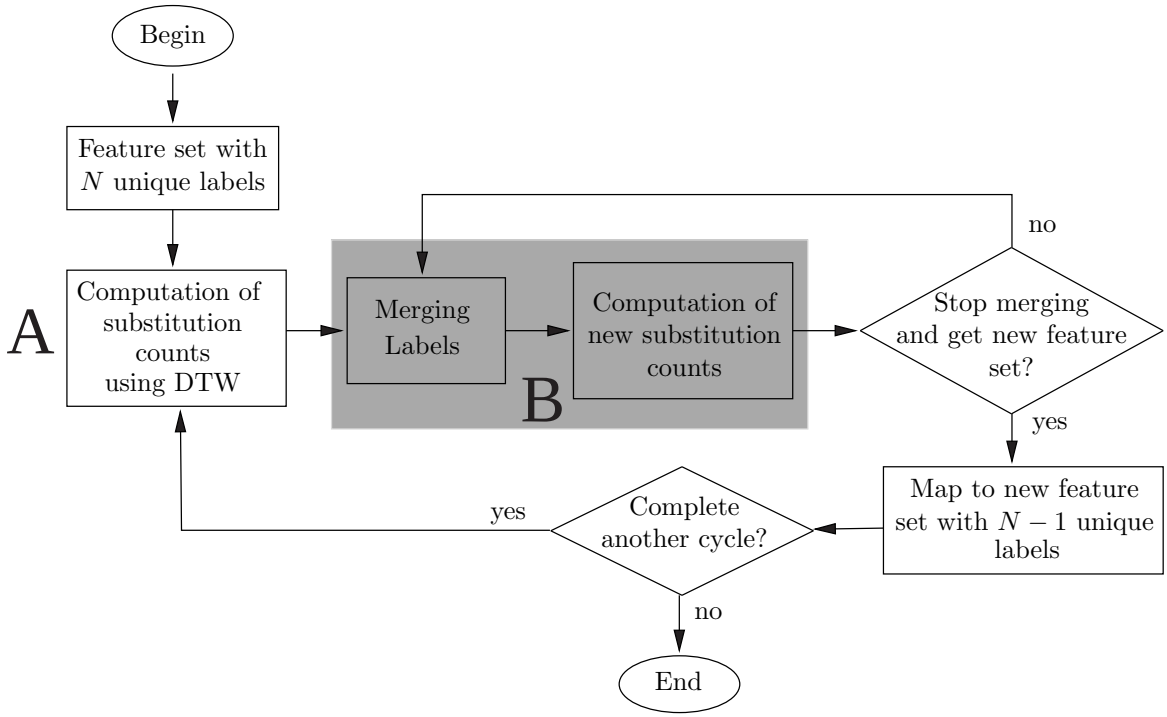


Figure 4.2: *A flowchart of the system describing the merging of labels, the computation of substitution counts and the generation of new merged feature sets.*

4.2.2 Using Prior Models to Smooth Other Models

Dirichlet Estimator

The Dirichlet estimator is a type of *maximum a posteriori* (MAP) estimator. The standard *a posteriori* (posterior) probability is [43]

$$P(M|D) = \frac{P(D|M)P(M)}{P(D)} \tag{4.4}$$

where M is the model and D is the data. The *maximum likelihood* (ML) estimations seek the model M that maximises $P(D|M)$. The MAP estimations seek the model M that maximises $P(M|D)$, i.e. $P(D|M)P(M)$ (as $P(D)$ is not a function of M). MAP becomes ML if the prior $P(M) = 1$.

For problems involving discrete PDFs, a natural (conjugate) prior to use is the Dirichlet prior [32]. The Dirichlet prior is attractive because it is easy to implement in the form of phantom counts. The new adjusted phantom (prior) count with the Dirichlet estimator

is:

$$c_i^* = c_i + \alpha * m_i \tag{4.5}$$

where m_i is a prior probability and α is a prior factor that is positive and real. The new adjusted $i-th$ probability is

$$p_i^* = \frac{c_i^*}{\sum_i (\alpha m_i + c_i)} = \frac{c_i^*}{\alpha + \sum_i c_i} \tag{4.6}$$

and $\sum_i m_i = 1$. Note that the phantom and observation counts in Equation 4.5 are those in a given state of the HMM. This estimator is used to smooth the transition probabilities of the HMM. This is implemented in two ways: using uniform prior probabilities, and using prior probabilities calculated from another well-trained model. The prior probabilities remain unchanged throughout every iteration of a specific training process.

Prior Models

By making use of the Dirichlet estimator the transition probabilities of one model can be smoothed with the transition probabilities of another well-trained model (the prior model).

The transition probabilities $\mathbf{A} = [a_{ij}]$ of a first-order model can be used as prior probabilities to smooth the transition probabilities $\mathbf{A} = [a_{kl}]$ of another first-order model for all $i = k$, and $j = l$ [1]. The transition probabilities $\mathbf{A} = [a_{ij}]$ of the first-order HMM are used as prior probabilities to smooth $\mathbf{A} = [a_{hij}]$ of the second-order model for all $\forall h$.

4.3 Experimental Approach

The same basic model structure of Chapter 3 (discussed in Section 3.5.3) is used for the experiments of this chapter.

The following experiments are conducted to explore possible routes to address the problem of data scarcity. The method of each experiment is to be discussed in detail. Since the aim of the experiments is to develop an idea of which solutions help improve the problem of data scarcity, a short conclusion of the results will be given. Our decisions will be evaluated on the equal error rates of the experiments performed. Section 3.6.3

[1] states i and k, and j and l are associated with the same label

showed that initialisation of the transition probabilities using BG and EP (refer to Section 3.5.4) produces similar results. The experiments in this chapter sometimes use BG, and sometimes EP.

The following approaches are taken to address data scarcity:

- Experiments are conducted using several merged feature sets consisting of 39 down to 9 phoneme labels, decreasing the number of labels in the feature set in steps of 5. The experiments with merged labels are compared to the experiment that uses the original feature set with 39 phoneme labels and the effects of using fewer parameters are investigated.
- Dirichlet estimation is used for smoothing purposes. This is carried out using uniform prior probabilities and using the transition probabilities of well-trained models as prior probabilities. The latter is divided into three categories:
  1. Using the UBM as prior model to smooth TSMs.
  2. Using first-order models as prior models to smooth second-order models.
  3. Using a merged model as prior model.
- Experiments are conducted where labels are merged and combined with smoothing techniques either by using uniform prior probabilities or by using transition probabilities of other models.

4.3.1 Merging Labels

The method described in Section 4.2.1 is used to merge labels in order to obtain fewer parameters of the HMM. Several merged feature sets are computed and used for speaker recognition. Labels are merged and these are used for both first-order and second-order experiments. A UBM, target models and impostor models are trained, using several merged feature sets (containing merged phoneme labels) and an original set containing 39 phoneme labels. For instance if a merged feature set containing 34 phoneme labels ($PG34$) is used for training, the following is done:

- A UBM is trained using PG34. This UBM is used to initialise target models and impostor models in the same way (Section 3.5.4). The target and impostor models are also trained using PG34.

- A matrix of SEP is calculated using the phoneme labels in PG34 and is used to initialise the PDF of all the speaker models (Section 3.5.4).

4.3.2 Uniform Prior Probabilities

With the uniform smoothing of the HMM transition probabilities, we add uniform prior (phantom) probabilities to observation counts. This means that the prior probabilities, m, in Equation 4.5 are all equal.

4.3.3 UBM Prior Models and Higher-Order Smoothing

With this type of smoothing the transition probabilities of a prior model are used as prior probabilities and are added to observation counts. (Refer to Equation 4.5). The more training data (observation counts) one has for the model that is being smoothed, the less influence the prior factor will have on the training process.

Combining smoothing with UBM prior models and higher-order (up to third-order) smoothing, the following possible paths can be explored, (shown in Figure 4.3). At the first (top) level, there are first-, second- and third-order UBMs. At the second level, there are first-, second- and third-order *target speaker models* (TSMs). One can use a UBM to smooth a TSM. In Figure 4.3 this is illustrated by arrows pointing down.

Higher-order models can be mapped to an equivalent first-order model using the ORED algorithm [12]. Second-order smoothing, using first-order prior models, is done as described in Section 4.2.2. Third-order smoothing, using second-order prior models, is done in much the same way as second-order smoothing, using first-order prior models. Higher-order smoothing is illustrated in Figure 4.3 by arrows pointing to the right. The diagonal arrows illustrate a combination of UBM prior models and higher-order smoothing.

In this chapter path c is used for first-order TSM smoothing, and path h is used for second-order TSM smoothing, as shown in Figure 4.3.

4.3.4 Smoothing using Merged Models as Prior Models

This approach is extremely complicated and proved to be of little significant value in speaker recognition. The concepts, approaches and experiments are found in Section A.5.

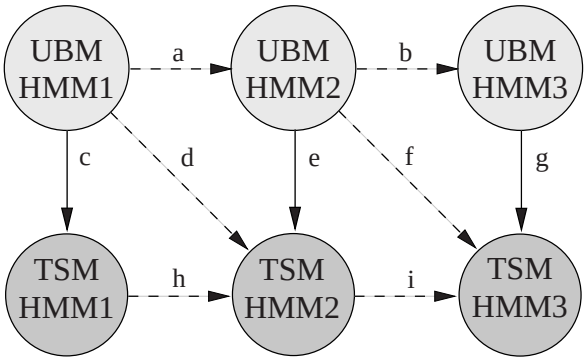


Figure 4.3: *Example of possible paths for UBM prior models to smooth target speaker models (TSMs) and higher-order smoothing up to third-order HMMs. Arrows represent the smoothing of TSMs using UBM prior models and dashed lines represent smoothing from a lower-order model.*

4.3.5 Merging Labels in Combination with other Smoothing Techniques

In this approach, labels of the original feature set are merged and are used to train merged models. During training of the merged models, a smoothing technique is used to estimate the merged model. The following smoothing techniques are used for the training of merged models:

- Uniform prior probabilities are used to smooth first-order and second-order merged models.
- Other models are used as prior models to smooth first-order and second-order merged models. UBM prior models are used to smooth first-order target models, while first-order target models are used to smooth second-order target models.

4.4 Experimental Results

A set of 40 first-order impostor models is trained without smoothing as described in Section 3.5.5. This set is used in all of the following experiments (both first-order and

second-order), unless stated otherwise. This is done to simplify verification and because the use of second-order impostor models is time-consuming[2]. The impostor models and target speakers are both initialised in the same way (Section 3.5.4).

4.4.1 Merging Labels

Several merged feature sets are used to train merged models as described in Section 4.3.1. The following feature sets and notations are used for the merged feature sets:

- Original - the original feature set with 39 unique phoneme labels.
- PGn - a feature set with n unique phoneme labels, with $n = 34, 29, 24 \ldots 9$.

The effect of merging labels is investigated in both first-order and second-order experiments. The main purpose of these experiments is to determine whether the use of merged feature sets can help improve speaker verification results. The effects of using merged feature sets that contain too few categories, are also studied.

First-Order Results

X1(BG): EER			
PGn	**4 CS**	**8 CS**	**16 CS**
Original	14.26%	11.23%	9.30%
PG34	15.66%	11.63%	11.35%
PG29	16.70%	12.85%	11.72%
PG24	18.13%	14.14%	12.55%
PG19	17.06%	13.98%	13.16%
PG14	20.03%	14.56%	14.88%
PG9	17.06%	22.62%	21.50%

Table 4.1: ***First-Order (X1)*** *EERs of the different merged feature sets. Initialisation with* ***normalised bigram counts (BG)****.*

Table 4.1 shows the EERs of first-order (X1) results, using initialisation with BG.

[2]Keep in mind that the choice of impostor models might influence the results. If the impostor models are trained exactly in the same fashion as the target models, the results might differ from when first-order, unsmoothed impostor models are used, as was done here.

Conclusion

As the number of categories in the feature set becomes too broad, the performance weakens. Specificity is lost using these feature sets and it becomes difficult to distinguish among speakers. PG9 obtains EERs that are approximately double the EERs when using the original feature set. The first-order results when using merged feature sets are not an improvement of the first-order results where the original feature set is used. The best results are obtained by using the original feature set with 39 phoneme labels. (Table A.1 shows similar results using EP.)

Second-Order Results

X2(BG): EER			
PGn	**4 CS**	**8 CS**	**16 CS**
Original	14.24%	13.99 %	11.62 %
PG34	18.03%	13.34 %	9.85 %
PG29	21.44%	14.63 %	12.18 %
PG24	21.46%	16.01 %	13.48 %
PG19	21.06%	15.60 %	14.41 %
PG14	19.75%	17.53 %	15.34 %
PG9	25.01%	24.26 %	21.56 %

Table 4.2: ***Second-Order (X2)*** *Equal Error Rates (EER) of the different merged feature sets. Initialisation with* ***normalised bigram counts (BG)****.*

Table 4.2 shows the EERs of second-order (X2) results, using initialisation with BG.

Conclusion

The second-order results also show a deterioration as the number of categories becomes too broad in the feature set. The best results are obtained by using a merged feature set containing 34 phoneme labels, and not by using the original feature set. (Table A.2 shows similar results using EP.) This is an interesting result, since it suggests that the merging

of labels can help in cases where data is scarce (such as in second-order experiments). It would be suggested that when a merged feature set is computed, the labels should be merged 2 at a time, once (and not 5 times as we do). The feature set with the least number of features should contain about 36 phoneme labels, otherwise categories become too broad.

4.4.2 Smoothing with Uniform Prior Probabilities

The purpose of these experiments is to determine whether smoothing with uniform prior probabilities gives better results than when smoothing is not used[3]. The original feature set with 39 phoneme labels is used for the experiments. Tables 4.3 and 4.4 show the EER results of first-order and second-order models with uniform prior probabilities added to the observation counts. In order to smooth conservatively (not too heavily nor lightly), a prior factor of 5 is used[4]. The first-order models are initialised using BG, and the second-order models are initialised using EP. (The aim of the experiment is not to compare initialisation methods, although initialisation using EP and BG produces similar results as can be seen in Section 3.6.3).

Conclusion

From the first-order results of Table 4.3, it seems that smoothing by means of using uniform prior probabilities does not improve the first-order results with no smoothing. In fact, the EERs of the first-order results increase when uniform smoothing is used. The EER of the second-order result with uniform smoothing is lower than the EER of second-order result without smoothing when using 16 conversation sides for training (16 CS). (See Table 4.4). On the other hand, the EER of the second-order result with smoothing is higher than the EER of the second-order result with no smoothing when using 4 conversation sides for training (4 CS). The less training data there is, the more influence the prior factor has on training and smoothing can become too heavily. Choosing uniform prior probabilities does not seem the best way of handling data scarcity. Other possibilities of prior probabilities need to be explored.

---

[3]Remember that first-order, unsmoothed impostor models are used for these experiments, as stated in the beginning of Section 4.4. If the T-norm impostors are also smoothed as is done with the target speaker models, the results might differ from those obtained in these experiments.

[4]Other prior factors were used in the experiments as well. Choosing the prior factor as 5 produced adequate results.

First-Order EERs Uniform Smoothing vs No Smoothing X1_UP(BG) X1(BG)			
Model	**4 CS**	**8 CS**	**16 CS**
X1_UP(BG)	15.01%	11.07 %	9.58 %
X1(BG)	14.26%	11.23%	9.30%

Table 4.3: *Smoothing first-order target models with uniform prior probabilities (X1_UP) (using a prior factor of 5) vs no smoothing of first-order target models (X1). Initialisation is done with* ***normalised bigram counts (BG)***

Second-Order EERs Uniform Smoothing vs No Smoothing X2_UP(EP) X2(EP)			
Model	**4 CS**	**8 CS**	**16 CS**
X2_UP(EP)	15.80%	14.15 %	11.25 %
X2(EP)	14.53%	14.95%	12.18%

Table 4.4: *Smoothing second-order target models with uniform prior probabilities (X2_UP) (using a prior factor of 5) vs no smoothing of second-order target models (X2). Initialisation is done with* ***equal probabilities (EP)***

4.4.3 UBM Prior Models and Higher-Order Model Smoothing

Section 4.3.3 mentions that the more training data (observation counts) we have for a model, the less influence the prior factor will have on the training process of that model. Consequently, by using the same prior factor, models that are trained using 4 conversation sides are more influenced by the prior model than models using 16 conversation sides for training.

The influence of the prior factor on the observation counts can be measured by $\frac{\alpha}{\sum_i c_i}$ in Equation 4.6. To keep the influence of the prior factor on the observation counts relatively low (smoothing too much), but not too low (smoothing too little), we choose to compare

prior factors of $\alpha = 10$, 20 and 40 [5]. (Typical ranges of the total number of observation counts in an HMM state are shown in Table A.11.)

Dirichlet Smoothing of First-Order Target Models

X1_DP(EP) : EER			
Prior Factor	**4 CS**	**8 CS**	**16 CS**
$\alpha = 0$	13.46%	10.99 %	9.58 %
$\alpha = 10$	14.10%	11.15 %	9.85 %
$\alpha = 20$	13.71%	10.75 %	9.77 %
$\alpha = 40$	13.50%	11.47 %	10.34 %

Table 4.5: *The EERs of experiments using the transition probabilities of a first-order UBM as prior probabilities to smooth the transition probabilities of first-order speaker models(X1_DP). Initialisation with* ***equal probabilities (EP)***

Table 4.5 shows the EERs obtained from using a first-order UBM to smooth first-order speaker models. (Taking path c in Figure 4.3). The results when using Dirichlet prior probabilities are compared to one where no smoothing is used ($\alpha = 0$).

Dirichlet Smoothing of First-Order Target Models: Conclusion

From the results in Table 4.5, it seems that using the UBM as prior model for target speaker models performs similar to first-order results with no smoothing ($\alpha = 0$).

Dirichlet Smoothing of Second-Order Target Models

Table 4.6 shows the results when smoothing the second-order speaker models using transition probabilities of first-order speaker models. (Path h in Figure 4.3). A prior factor of $\alpha = 10, 20$ and 40 is used. These results are compared to the results where no smoothing is used ($\alpha = 0$).

[5] Through practical observation the ratio of prior factor and observation counts $\frac{\alpha}{\sum_i c_i} < 1/3$ for most of the speaker models, using 4 conversation sides for training.

X2_DP(EP) : EER			
Prior Factor	**4 CS**	**8 CS**	**16 CS**
$\alpha = 0$	14.53%	14.95 %	12.18 %
$\alpha = 10$	13.35%	11.88 %	9.40 %
$\alpha = 20$	12.78%	11.80 %	10.88 %
$\alpha = 40$	13.58%	11.24 %	8.37 %

Table 4.6: *The EERs of experiments using first-order target models to smooth second-order target models(X2_DP). Initialisation with* ***equal probabilities (EP)***

The results when using prior factors of 10, 20 and 40 perform similarly using 4 and 8 conversation sides for training. The results where Dirichlet smoothing is used, are all improvements of the results where no smoothing is used ($\alpha = 0$). Using a prior factor of $\alpha = 40$ gives the best EER using 16 conversation sides for training. The second-order results using a prior factor of $\alpha = 40$ seem to be a marked improvement on the results where there is not smoothed. It seems therefore that first-order target models are a good choice as prior models to smooth second-order target models.

Figure 4.4 shows the comparison of two results:

- Uniform prior probabilities are used to smooth second-order target models (marked "Uniform" on the figure).
- First-order target speaker models are used to Dirichlet smooth second-order target speaker models (marked "Dirichlet" on the figure). A prior factor of $\alpha = 40$ is used.

The experiment shows that using first-order TSMs for smoothing second-order TSMs performs far better than using uniform prior probabilities, especially when using 8 and 16 conversation sides for training.

We also experiment using a second-order UBM to smooth second-order speaker models. (Path e in Figure 4.3). We use an $\alpha = 40$. This gives EER results of 21.12% (4 conversation sides), 14.95% (8 conversation sides) and 9.86% (16 conversation sides), using the original feature set. Comparing these results to the results in Table 4.6, one sees that these results fare worse especially in experiments that use a small number of conversation sides (4 and 8). The UBM carries no speaker-specific information. Using it directly for

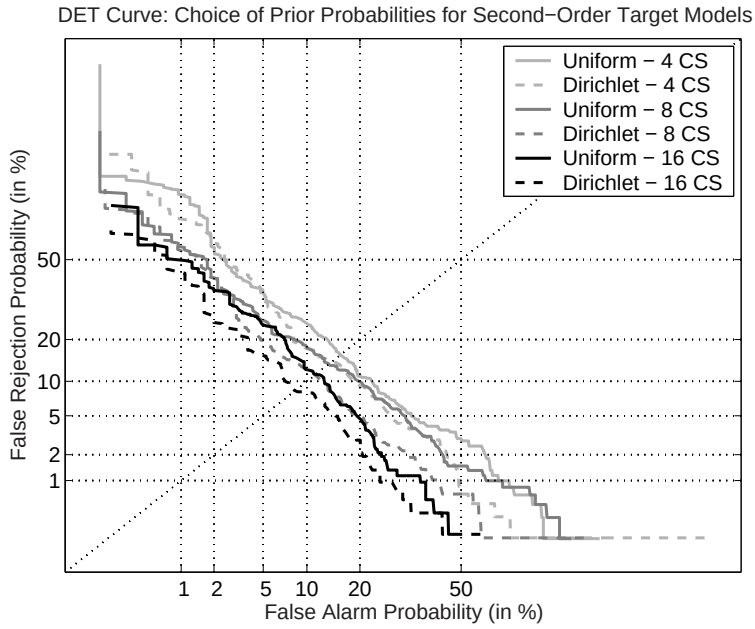


Figure 4.4: *DET curves of experiments using first-order TSMs as prior models to smooth second-order TSMs ("Dirichlet") compared to experiments where smoothing is done using uniform prior probabilities("Uniform"). Initialisation is done using EP.*

smoothing second-order TSMs is not as good a choice as using first-order TSMs (with speaker-specific information).

Bear in mind that by smoothing a second-order model with a first-order model, we so to speak structure a mixture between these two models. The more training data available for the second-order training, the less will be the influence of the first-order model. It therefore makes sense to compare the results achieved by using first-order models to smooth second-order models to those attained solely by first-order models. In Chapter 5 a thorough comparison of these latter two experiments are made by making use of DET curves and significance tests.

Dirichlet Smoothing of Second-Order Target Models: Conclusion

The best second-order results are obtained by using the first-order target models as prior models to smooth the second-order target models.

4.4.4 Merging Labels in Combination with other Smoothing Techniques

The combination of using merged labels with the smoothing techniques described in Section 4.3.5 does not show any improvement if compared to the best results obtained by using the original feature set containing 39 phoneme labels. The results of using a combination of the merging of labels and uniform or Dirichlet smoothing can be found in Tables A.8 to A.10.

4.4.5 Summary

Several configurations have been discussed: merging features to create fewer parameters, the choices of prior probabilities (uniform or the transition probabilities of prior models). When labels are merged as in the experiments of Section 4.4.1, the performance of the first-order systems becomes worse. The performance of the second-order systems (using 16 conversation sides) improves slightly when using a merged feature set with 34 phoneme labels (PG34) compared to the original feature set containing 39 phoneme labels (Table 4.2). However, this merged second-order result (EER = 9.85%) is not an improvement of the EER result obtained when first-order target models are used to smooth second-order target models (EER = 8.37%, using the original feature set and 16 conversation sides). (See Tables 4.2 and 4.6.) Table 4.7 is a summary of viable ideas that surfaced

Inadequate Idea	**Promising Idea**
Smoothing using uniform prior probabilities	
Merging features to decrease parameters	Dirichlet smoothing by using first-order speaker models to smooth second-order speaker models
Using unmerged equivalent models as prior models	

Table 4.7: *Summary of the ideas that surfaced exploring several configurations of smoothing models and merging phoneme labels*

during the experiments carried out in this chapter [6].

In Chapter 5 the ideas in Table 4.7 are incorporated to design a smoothing configuration for phonetic speaker recognition. An in-depth study of the results are done, in which they are evaluated using DET curves and significance tests. Since there is little difference between using BG and EP to initialise transition probabilities, it does not really matter which method we choose. In all the experiments to follow in this study, we choose to initialise transition probabilities using BG.

[6]These ideas are based on the way they were implemented in this study

Chapter 5

Significance Tests for Second-Order Experiments

5.1 Introduction

In Chapter 3 we learn that modelling of the SEP of the phoneme recogniser can improve the phonetic speaker recognition system a great deal. Chapter 4 deals with the problem of data scarcity and a few possible solutions are investigated. A Dirichlet estimator is used to smooth the transition probabilities of the HMM.

It seems that the more sensible way of using this estimator for smoothing would be to use first-order models as prior models to smooth second-order models. It also seems sensible to choose the prior factor of this estimator proportional to the number of destination links of the particular state in the HMM. In this chapter we incorporate the above approach for Dirichlet estimation in order to perform first-order and second-order phonetic speaker recognition. The same basic model structure as in the previous two chapters are used (Section 3.5), and the substitution errors of the phoneme recogniser are modelled in the PDFs of the HMMs.

In the previous chapter an idea was broadly formed on which approaches seemed promising to handle data scarcity. Subsequently, the results of these approaches were not compared in detail. This chapter focuses more fully on the significant differences between two approaches, while making use of common training and testing data. First-order and second-order experiments are compared by means of the DET curve (Section 2.3.2). A thorough investigation of these curves are done, using a significance test to evaluate these curves and to tell whether the improvement is significant.

5.2 Significance Tests

The evaluations of the experiments in this chapter make use of a significance test called the McNemar's test. This test can be used to compare two algorithms, $A1$ and $A2$, that are both evaluated on the same data set. The following notation and definitions are taken in parts from [16].

The joint performance of the two algorithms, $A1$ and $A2$, can be summarised in a 2×2 table as follows:

		A2	
		Correct	Incorrect
A1	Correct	N_{00}	N_{01}
	Incorrect	N_{10}	N_{11}

where:

N_{00} = Number of utterances which $A1$ classifies correctly and $A2$ classifies correctly
N_{01} = Number of utterances which $A1$ classifies correctly and $A2$ classifies incorrectly
N_{10} = Number of utterances which $A1$ classifies incorrectly and $A2$ classifies correctly
N_{11} = Number of utterances which $A1$ classifies incorrectly and $A2$ classifies incorrectly

Let the true (but unknown) error rates of $A1$ and $A2$ be $p1$ and $p2$ respectively. By analogy of N_{ij}'s, define q_{ij}'s:
$q_{00} = \Pr(A_1 \text{ classifies correctly})$ etc., where $\{u_i\} = u_1, u_2, \ldots, u_n$ is a sequence of labelled utterances for recognition, and $n = N_{00} + N_{01} + N_{10} + N_{11}$. Note that $p1 = q_{10} + q_{11}$, and $p2 = q_{01} + q_{11}$.

We would like to test the null hypothesis:

$$\mathbf{H_0} : p_1 = p_2 = p$$

$\mathbf{H_0}$ being equivalent to $\mathbf{H_0^1} : q_{01} = q_{10}$.

Defining $q = q_{10}/(q_{01} + q_{10})$, a further null hypothesis is $H_0^2 : q = \frac{1}{2}$. The parameter q represents the conditional probability that $A1$ will make an error on an utterance, given

that only one of the two algorithms makes an error. The latter hypothesis is based on the assertion that, given that only one of the algorithms makes an error, it is equally likely to be either $A1$ or $A2$.

One therefore needs to examine only the utterances where only one of the two algorithms makes an error, in other words N_{10} and N_{01}. Let $k = N_{10} + N_{01}$. We test the null hypothesis by applying a two-tailed test, using the following binomial distribution:

$$P = \left\{ \begin{array}{ll} 2 \sum_{m=N_{10}}^{k} \binom{k}{m} \left(\frac{1}{2}\right)^k & \text{when } N_{10} > k/2 \\ \\ 2 \sum_{m=0}^{N_{10}} \binom{k}{m} \left(\frac{1}{2}\right)^k & \text{when } N_{10} < k/2 \end{array} \right\}$$

$\mathbf{H_0}$ is rejected when P is less than some significance level ϵ. Typical values of ϵ are 0.05, 0.01 or 0.001. If k is large enough ($k > 50$), and n_{10} is not too close to k or 0, a Normal approximation of the binomial probability may be used, with the expected value $E(N_{10}) = k/2$ and a variance $Var(N_{10}) = k/4$.

5.3 Experimental Approach

5.3.1 Newly Defined Dirichlet Estimator

During standard maximum likelihood estimation (using no smoothing), the transition probabilities of certain links in the HMM that were not observed become zero. These links are then permanently removed and cannot reappear in the next training cycle. The removal of links can be desirable because the HMM can otherwise be too large and becomes impractical to apply. (The training of a large HMM can be time-consuming.) With Dirichlet estimation phantom counts are added to observation counts in a specific state of the HMM. If the phantom counts are all non-zero in a specific state, the transition probabilities would also be non-zero in that state. The transition links of that state would therefore not disappear during training. On the other hand, if too many links of the HMM disappear during training, it is a sign of data scarcity. In such a case it would be desirable to use a smoothing estimator such as the Dirichlet estimator.

To prevent the HMM from becoming too large (especially higher-order HMMs), a cutoff is defined, where if the number of training examples exceeds this cutoff, we revert to

standard maximum likelihood estimation.

$$\lambda = k_\lambda * linkno \qquad \text{where } k_\lambda > 0 \tag{5.1}$$

Typical values of k_λ are $k_\lambda > 3$, and $linkno$ is the number of destination links in a given state of the HMM. We also use a prior factor that is proportional to the number of destination links in a specific state of the HMM. A new prior factor is defined

$$\alpha = k_\alpha * linkno \qquad \text{where } k_\alpha > 0 \tag{5.2}$$

where typical values of k_α range from 0.25 to 1.

In the experiments in this chapter, this newly defined Dirichlet estimator is used, choosing parameters $k_\alpha = 1$ and $k_\lambda = 5$. Choosing $k_\alpha = 1$, corresponds to typical values of α ranging between 20 and 40 for an HMM with 40 states. An $\alpha = 20$ or 40 are used in the experiments of Section 4.4.3. The parameter k_λ is chosen relatively high in order to smooth conservatively.

5.3.2 Using the McNemar Test with DET curves

To generate DET curves, we make use of FAR's and FRR's, which are generated by evaluating T-Norm output scores of a test sequence at various thresholds, th (See Section 2.3.2). Consider two algorithms, $A1$ and $A2$. A set of FAR's and FRR's are generated as a function of the threshold, th, for each of the two algorithms. Let $th1(n)$ be the n-th threshold that generates FAR1(n) and FRR1(n) in $A1$. Let $r1(n)$ = FRR1(n)/FAR1(n) and let m be the index for which the ratio $r2(m)$ =FRR2(m)/FAR2(m) has the closest value to $r1(n)$[1].

The next step is to compute McNemar parameters N_{10} and N_{01} of the output scores of $A1(n)$ at $th1(n)$, and $A2(m)$ at $th2(m)$. This is done for every threshold in $A1$ and their corresponding thresholds in $A2$. Using Equation **??**, it is now possible to calculate significance probabilities P for each $A1(n)$ and corresponding $A2(m)$.

Using the definitions in Section 5.2, it is also possible to compute which output sequence performs best:
$N10 - N01 > 0$ means that algorithm $A2$ has performed better than algorithm $A1$. We now have two important outputs for every point on the DET curve of $A2$ and its corresponding point at $A1$:

[1]The procedure is not symmetrical and the answer may differ slightly if the algorithms $A1$ and $A2$ are interchanged

- One that states whether $A2$ has performed better than $A1$.
- One that states the significance probability P in Equation **??** of the difference between A2 and A1.

By using these two outputs, it is possible to select areas on the DET curve where A2 has performed better than A1 and to plot the significance probability P at these points.

5.3.3 Experimental Setup

The same set of (unsmoothed) first-order impostor models is used for verification as the set used in the previous chapters. This set of impostor models is used for verification in first-order and second-order experiments. The set of impostors is initialised with the UBM in the same way as the set of TSMs is.

Using path c in Figure 4.3, a set of first-order TSMs (X1) is trained that is smoothed and initialised with a UBM. A set of second-order TSMs (X2) is trained by smoothing their transition probabilities with the transition probabilities of the above-mentioned set of first-order TSMs. This is illustrated in Figure 4.3 by continuing with path h after having taken path c for the first-order models[2]. Using the method described in Section 5.3.2, significance probabilities are calculated for X1 and X2 (at various operating point on the DET curves). A significance level of of $\epsilon = 0.1$ is used.

5.4 Experiments using the McNemar Test and DET curves

Figure 5.1 shows the DET curves of the first-order (X1) and the second-order (X2) experiments using 16 conversation sides for training. The thicker lines on these DET curves indicate areas where there is a significant difference between the results of X1 and X2, while the thin lines indicate where there is not. The dotted radial curves show the ratio

[2]A significance test shows that using the UBM as prior model for first-order target speakers is not a significant improvement over using the first-order models without smoothing (see Section A.8.1). It suggests that there is sufficient training data for first-order speaker modelling. The first-order experiments of Chapter 5 and 7 use the UBM as prior model. Since this is not an improvement of the first-order results, it is recommended that the UBM is not used as prior model.

between false rejection rates and false alarm rates r = FRR/FAR at 100, 10, 1, 0.1 and 0.01. From Figure 5.1 it can be seen that the second-order result is not a significant improvement of the first-order result over the widest range of the DET curve. (Neither are the second-order results a significant improvement of first-order results when using 4 and 8 conversation sides for training. These results can be found in Figures A.19 and A.20. The probability levels, where X2 has performed better than X1, can be found in Figures A.21 to A.23.)

Comparing second-order results to first-order results seems to be a valid experiment to rerun when more data is available.

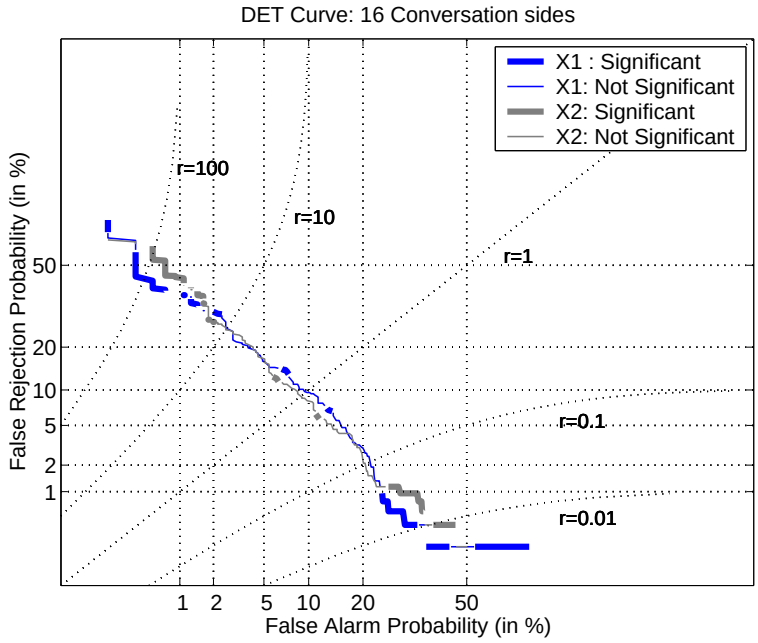


Figure 5.1: *DET curve of X1 and X2 using 16 conversation sides for training, indicating the significant difference with thicker curves*

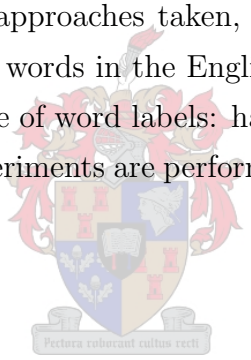
5.5 Summary

In this chapter we carried out experiments with a Dirichlet estimator, using a prior factor that is proportional to the number of destination links in a given state of the HMM. In the design of the Dirichlet estimator, we implemented a cutoff. If the number of training examples exceeds this cutoff, we revert to standard maximum likelihood estimation. The cutoff prevents the HMM from becoming too large and allows the HMM links to disappear

during training (the transition probabilities of those links become zero). A conservative cutoff is chosen that is high enough, so that at that value, data scarcity is not a problem and Dirichlet smoothing therefore unnecessary.

Second-order results using first-order prior models are compared to first-order results (using no smoothing). They are evaluated using the McNemar significance test and DET curves. Compared to the first-order results, the second-order results show no significant improvement over the widest range of the DET curves. A second-order model for target speakers is designed that at worst, would perform similarly to the first-order model[3]. Data scarcity is a problem, especially when modelling the speakers with second-order HMMs. (This can be seen when looking at the typical ranges of the total number of observation counts in a second-order HMM state in Table A.11). Were more data available, it would be worth re-evaluating the second-order experiments performed in this chapter.

This chapter concludes the experiments done with phonetic speaker recognition. In the next chapter we introduce the approaches taken, using word labels to perform speaker recognition. There are far more words in the English language than phonemes. We look at the issues surrounding the use of word labels: having so many words puts a limitation on the models one can use. Experiments are performed with particular selections of words as feature vectors.



[3]If there is not enough data available for the training of second-order models, the smoothing of a second-order speaker model with a first-order speaker model results in a model that is close to equivalent to the first-order model with which it was smoothed.

Chapter 6

A lexical approach to Speaker Recognition

6.1 Introduction

When one speaks, one tends to have certain habits regarding specific words or word patterns such as "okay, yes" or "uh". These lexical habits could stem from the social groups with whom we associate, or could be individual habits picked up with time. It is easier to distinguish the speech of familiar speakers [11], as we are accustomed to their speech idiosyncrasies. In this chapter we endeavour to use a lexical approach to do speaker recognition, i.e. to use word labels as feature set to train speaker models. Since there are so many words, a simple approach is followed by counting single word frequencies (the number of times an individual word is used by a speaker).

There are two ways of handling lexical speaker recognition: The first is to use a non-speaker-specific approach, where the same, general selection of words is used to train models for all the speakers. In this approach, words that are used frequently by the average speaker are used as feature set, while the rest of the words are ignored. This selection is then varied by setting a threshold on the number of times a word must be used to be included in the feature set. The second way is to use different selections of words, so that each speaker model is trained with its own speaker-specific feature set. This approach does not work as well, since the selection of words in the feature set is too small owing to data scarcity.

6.2 Background

To understand the use of words for speaker recognition more fully, let us consider the typical conversations amongst a group of people and person X. Looking at the word usage of person X, there are a few possibilities to consider:

1. There are some words that everyone uses frequently, such as the word "I".
2. Some words person X would use often, and other people almost never.
3. Some words person X would never use, while other people use them fairly regularly.
4. Some words are used rarely by everyone.

These types of word usage can be investigated in speaker recognition.

6.3 Speaker Modelling

6.3.1 Databases and Handset Labels

During the 2001 and 2002 evaluation period, NIST provided participants with a set of automatically transcribed words. The labelling was done by analysis and classification (implementing a word spotter) of the speech signal in Switchboard I and Switchboard II. Switchboard I also contains a set of genuine transcriptions. The substitution error rate of the word classifications varies, depending on the number of words included in the feature sets. Typical error rates range from as much as 35% to 25%[1]. The Switchboard I corpus has almost 27 000 genuine words and 24 000 classified words in its dataset. The Switchboard II corpus has approximately 23 000 classified words in its dataset.

[1]The fewer words in the feature set, the fewer words there are to cause confusion and hence lower substitution error rates).

6.3.2 Model Structure

The type of model for the speakers should now be considered. A principal matter that needs to be kept in mind is the wide range of words at hand. The model chosen for phonetic speaker recognition is an HMM as we are interested in the time-dependencies among the phoneme labels. With word labels as feature set, choosing an HMM as speaker model is unwise, considering that there are approximately 27 000 word labels in the Switchboard II corpus. This type of modelling would cause speed and memory problems.

Given the huge number of word labels, we settle for the simplest approach: The word frequencies (the number of times the word appears in a training sequence) are counted and normalised to obtain word probabilities. This type of model structure applies to all the speakers. The speakers include UBM speakers, target speakers and impostor speakers.

UBM Speakers

Speakers from jack_1 (jackknife set 1) to jack_9 in Switchboard II are selected as the UBM speakers, as was also done for the phonetic speaker recognition. The classified label counts of the UBM are smoothed by adding a uniform count of 1 divided by the number of words in the feature set to all the classified label counts and to an extra component that is referred to as the garbage component or label. This garbage label represents all other words not included in the feature set. It is important to note that the garbage component is not included in the actual counting of words, for it would weigh too much. Effectively the garbage component is zero before training, and after smoothing becomes non-zero.

Target Speakers

The target speakers are selected from jack_0 (Switchboard II). The classified label probabilities of the UBM are used as priors to smooth the target speakers in most of the experiments in this chapter. In Section 4.3.3 it states that, the more training data (observation counts) one has for a certain label, the less influence the prior factor will have on smoothing. There are fewer observation counts per label using words than there are per label using phonemes, since there are so many more words than phonemes. Taking the aforesaid into consideration, one should not use too big a prior factor in smoothing the target speakers with the UBM.

The prior factor is chosen to be proportional to the number of words in the feature set. In order for the UBM not to carry too much weight when using it for smoothing of the target speaker models, a prior factor of 0.1 times the number of words in the feature set is chosen. This means, using the data of Switchboard II, that for a feature set containing less than 5 000 words, $\frac{\alpha}{\sum_i c_i} < 1/3$ (See Equation 4.6).

Impostor Speakers

A set of 80 impostor speakers is chosen from Switchboard I. These impostor speakers are smoothed in exactly the same way as is done with the target speakers, using a prior factor of 0.1 times the number of words in the feature set.

Feature Set

The selection of words chosen for the feature sets all come from words included in Switchboard I. We explore several selections of these words, experimenting both with speaker-specific feature sets and general feature sets that are not speaker-specific.

6.4 Selection of Word Labels as Feature Set

6.4.1 Selection based upon word count

The word labels in Switchboard I and II can be counted. The number of times a word appears in the dataset is referred to as the *word count*. The classified word labels of Switchboard I are used as feature sets.

It is highly unlikely that a word with a low word count in the Switchboard I corpus appears in the training set or testing set of Switchboard II. Also, words with a low word count are most likely to be topic-specific. (NIST Switchboard are telephone conversations between two people who have been given a topic on which to talk). Bearing this in mind, if some words need to be eliminated from the feature set, one can eliminate those with a low word count and select a group of words with a word count greater than a given threshold. We also investigate what the effect of eliminating words with a high word count from the feature set has on speaker recognition.

6.4.2 Selection of Words Based upon Speaker Entropy

The (conditional) *speaker entropy* (in bits per symbol) of a word w_n is defined as [11]

$$H(S|w_n) = -\sum_i P(s_i|w_n) \log_2 \left[P(s_i|w_n)\right], \tag{6.1}$$

where s_i is the i-th speaker out of a selected group of speakers S, and $P(s_i|w_n)$ is estimated by

$$P(s_i|w_n) = N_{s_i}(w_n)/N(w_n) \tag{6.2}$$

and $N_{s_i}(w_n)$ is the number of times word w_n is used by speaker i, and $N(w_n)$ is the number of times w_n is used by all the speakers in the selected group.

Entropy is also referred to as uncertainty [49]. A word with low speaker entropy can be used to discriminate between speakers, as it originated with high certainty (low uncertainty) from specific speakers. Such words indicate a high certainty that specific speakers use the words either significantly more than other speakers or significantly less than other speakers. Unfortunately, a word that occurs only once in the training set will have the lowest possible speaker entropy, without necessarily improving speaker discrimination. This is because the word might not appear in the test set, or its statistics may differ drastically when more data is collected.

A better approach for selecting words for speaker recognition is to combine speaker entropy with the frequency of the words in the training set. Good words for speaker discrimination should have a low entropy combined with a high occurrence. This approach selects a general set of words as feature set. (The feature sets used to train each target model are the same.)

6.4.3 Selection of Words Based upon Log-Probabilities

The UBM is modelling the probabilities of word usages of the average speaker. If it is possible to find the usage of words from the target speakers that deviates from the usage of words in the average model (UBM), one can then tell which selection of words would be valuable to model a specific target speaker. We do this by evaluating the following

deviation:

$$d(n,i) = \log[P(w_n|\text{UBM})] - \log[P(w_n|\text{TM}_i)] \tag{6.3}$$

where $P(w_n|\text{UBM})$ is the probability of the n-th word (w_n) in the UBM, and $P(w_n|\text{TM}_i)$ is the probability of the n-th word (w_n) in the i-th target model (TM_i). The closer the deviation is to zero, the less difference there is between the use of w_n of the average speaker and the speaker of TM_i . A large positive deviation indicates that the speaker of TM_i uses w_n significantly more than the average speaker. A large negative deviation means that the speaker of TM_i uses w_n significantly less than the average speaker. According to this scheme, words with large $|d(n,i)|$ are selected for use with TM_i. This approach selects different feature sets to train each target speaker (target-specific).

6.5 Training Genuine Word Labels : Approach

With the phonetic speaker recognition component of this study, the SEP of the phoneme recogniser is accounted for by modelling it in the PDF of the HMM. In the training of classified word labels, the number of times a specific word is detected (classified) is counted. These counts are normalised over all words to give the probabilities of all the classified words. In this process, the errors of the word recogniser is not accounted for in the probabilities. We could make use of the substitution errors of the word recogniser to model the word recogniser errors. For instance, from the substitution errors, it is possible to calculate the probabilities of genuine word labels, given the classified word label. We refer to these probabilities as reversed substitution probabilities. (Substitution probabilities are the probabilities of a classified word label, given a genuine word label. Refer to Section 3.4.3).

Let the probabilities of the genuine word ‘dumb’ be 75%, and the genuine word ‘down’ be 25%, given that the word ‘dumb’ is classified. The reversed substitution probabilities are:

$$P(\text{genuinely ‘dumb’} \mid \text{‘dumb’ is classified}) = 0.75$$

$$P(\text{genuinely ‘down’} \mid \text{‘dumb’ is classified}) = 0.25$$

If the classified word ‘dumb’ is detected four times, then in the training process, the genuine word ‘dumb’ is counted 3 times and the genuine word ‘down’ is counted once. In this way the genuine word labels are trained. During the training of genuine labels a smoothing estimator can be used.

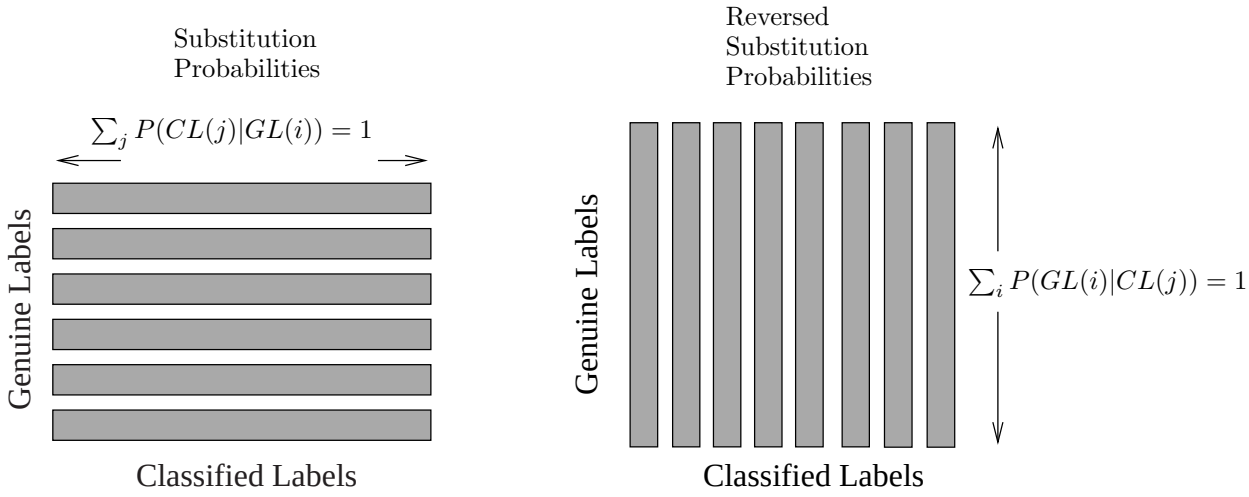


Figure 6.1: *The calculation of (a) substitution probabilities and (b) reversed substitution probabilities.*

Figure 6.1 shows the calculation of substitution probabilities $P(CL(j)|GL(i))$ and reversed substitution probabilities $P(GL(i)|CL(j))$, where $CL(j)$ is the j-th classified label and $GL(i)$ is the i-th genuine label. To generate substitution probabilities from substitution counts, the substitution counts are normalised over j (associated with the classified labels). In order to generate reversed substitution probabilities, the substitution counts are normalised over i (associated with genuine labels).

We incorporate the training of genuine word labels by using the following approach:

- At the input of the speaker model, we have classified word labels.
- We train the genuine word labels by using the reversed substitution probabilities, $P(GL|CL)$, and by counting the genuine labels. For each example found for a given classified word, $CL(j)$ (classified count increased with 1), the genuine counts are increased by the following equation:

$$N^*_{GL(i)} = N_{GL(i)} + P(GL(i)|CL(j)) \qquad \forall\, i \tag{6.4}$$

  where $N_{GL(i)}$ is the word count of genuine label GL(i) before training, and $N^*_{GL(i)}$ the word count after training.
- At the end of the training cycle, the counts of the genuine labels are smoothed by means of a Dirichlet estimator. The probabilities of the genuine labels are calculated.

These genuine probabilities, $P(GL)$, are converted to classified probabilities, $P(CL)$, by multiplication with the substitution probabilities, $P(CL|GL)$:

$$P(CL(j)) = P(CL(j)|GL(i))P(GL(i)) \tag{6.5}$$

6.6 Experiments

6.6.1 Selection of Words Based upon Minimum Word Counts

We experiment by varying the threshold for the minimum word count in the classified word set of Switchboard I. The effect of also excluding words with the highest word counts have been investigated and is found not to work as well as excluding words with a minimum word count. (The results can be found in Section B.2.)

A total of 5 feature sets are used for experiments by setting the minimum word count equal to 5, 20, 50, 100 and 400. These feature sets are then used to train classified word probabilities. First a UBM is trained from the set of UBM speakers (speakers from jack_1 to jack_9 of Switchboard II) as described in Section 6.3.2. The UBM is used to smooth the target speakers using a prior factor of 0.1 times the number of words in the feature set. The impostor speakers are trained and smoothed in exactly the same fashion as the target speakers.

Figure 6.2 shows the DET curves of experiments using the feature sets containing classified words, varying the minimum word count (using 4 conversation sides). Table 6.1 contains the EERs of these DET curves, using 4, 8 and 16 conversation sides for training. The overall accuracies are computed by including target models trained on 4, 8 and 16 conversation sides.

The evaluation of these experiments, as well as the rest of the experiments of this chapter, is done by classifying speakers as target speakers if the T-Norm score is greater than 1.28 (corresponding to a 10% expected false alarm rate). The overall accuracy is computed by adding the number of incorrect classifications and dividing it by the total number of trials. (See Section 2.3.1).

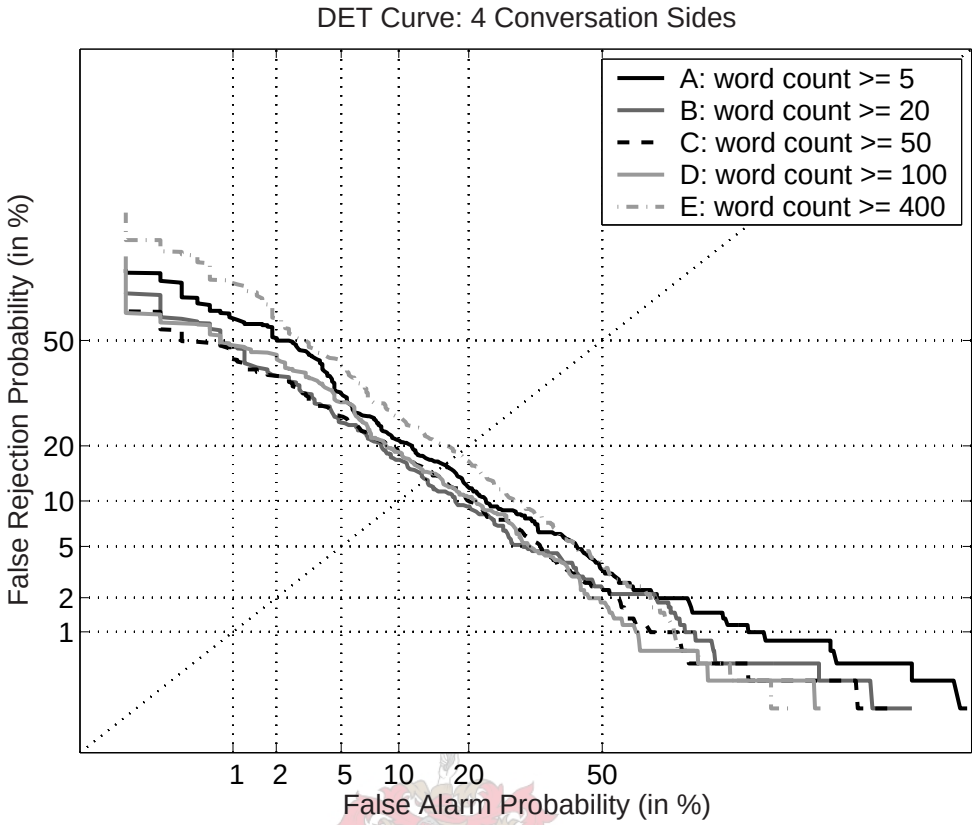


Figure 6.2: *DET curves of feature sets containing words that are selected based on minimum word counts, using 4 conversation sides for training.*

Feature Set	Equal Error Rate (EER)			Overall Accuracy
	4 CS	8 CS	16 CS	
A: word count >= 5	16.01 %	9.71 %	7.81 %	83.37 %
B: word count >= 20	13.35 %	9.87 %	7.81 %	88.94 %
C: word count >= 50	14.41 %	11.56 %	8.46 %	84.90 %
D: word count >= 100	14.19 %	10.99 %	9.66 %	80.23 %
E: word count >= 400	18.25 %	14.30 %	11.34 %	72.35 %

Table 6.1: *EER results of feature sets with word counts* $>= 5, 20, 50, 100$ *and* 400*, using 4, 8 and 16 conversation sides (CS) for training.*

Conclusion

Feature set B (with word counts greater than or equal to 20) obtains the best overall accuracy of 88.94%. As we increase the minimum word count of words included in the feature set, some words that have a value for speaker recognition are excluded from the feature set. The results show the gradual drop as the minimum word count is increased, using feature sets with a minimum word count of 50, 100 and 400. Feature set E (minimum word count of 400) dropped to an overall accuracy of 72.35%. The EER when using feature set A is higher than the EER when using feature set B using 4 conversation sides for training. Feature set A includes words that are irrelevant for speaker recognition. These irrelevant words are words with a relatively low word count (< 15) that have a low probability of appearing in all of the training, impostor and test sets, and are likely to be topic-specific. When using 8 and 16 conversation sides for training, there is barely any difference between the EERs of experiments using feature set A and feature set B. (This is also evident in the DET curves of experiments using 8 and 16 conversation sides for training shown in Figures B.1 and B.2). Thus the more training data there is, the less negative influence the irrelevant words have on speaker recognition performance.

Summary

Of the group of feature sets investigated, Feature set B (minimum word count of 20) seems to be the wisest selection of words for speaker recognition purposes, especially when using 4 conversation sides for training.

6.6.2 Selection of Words Based upon Entropy and Log-Probabilities

The approach that is followed when selecting words based upon word entropy and log-probabilities are explained in Sections 6.4.2 and 6.4.3. Because of data scarcity, these approaches do not work well. Experiments using a word selection based on entropy and log-probabilities are discussed in Sections B.3 and B.4.

6.6.3 Training Genuine Word Labels

These experiments are based on the approach described in Section 6.5. Selections of words in Switchboard I are used as feature sets. Substitution counts are generated using

the genuine and classified word labels in Switchboard I. We experiment using 2 groups of genuine and classified word labels. (Classified words are used as feature set for the training of both classified and genuine word probabilities):

FEAT1: The sets of genuine and classified words both have a minimum word count of 20.

FEAT2: The sets of genuine and classified words both have a minimum word count of 50.

Two sets of substitution counts, substitution probabilities and reversed substitution probabilities are computed, using the set of genuine and classified word labels belonging to FEAT1 and FEAT2. With the training of classified word labels, we use the UBM as prior model to smooth the target speakers, using a prior factor of 0.1. Equation 6.4 is used to train genuine word counts. Genuine word probabilities of a UBM are trained by using the classified feature sets and the reversed substitution probabilities belonging to FEAT1 and FEAT2 and are inserted into Equation 6.4. After the training cycle, the genuine word probabilities are converted to classified word probabilities with the aid of Equation 6.5.

The results of using classified words for training are compared to using genuine words for training, using both FEAT1 and FEAT2. The DET curves of the experiments using FEAT1 and 4 conversation sides for training are shown in Figure 6.3. TGL are the experiments training the genuine word labels and TCL are the experiments training the classified word labels.

Conclusion

The running time of TCL takes in the order of 1 minute while the running time of TGL could be as long as 5 minutes. TGL does not perform much better than TCL and taking into consideration the longer running time of TGL, TCL seems to be the better approach for training. (TGL and TCL perform similar for FEAT 1 and FEAT 2, using 4, 8 and 16 conversation sides. These results are shown in Figures B.11 to B.15.)

To our knowledge, it could be possible that the automatic label classification procedure of Switchboard I and Switchboard II differs and that the substitution errors of Switchboard I are mismatched with those of Switchboard II. (We did not have the genuine labelling for Switchboard II and could unfortunately not compute substitution errors for Switchboard II). The substitution errors could be dependent on the dataset, even if the labelling procedure of Switchboard I and II is the same. It could be that the computation of substitution errors for the word classifier is complex and needs a comprehensive dataset

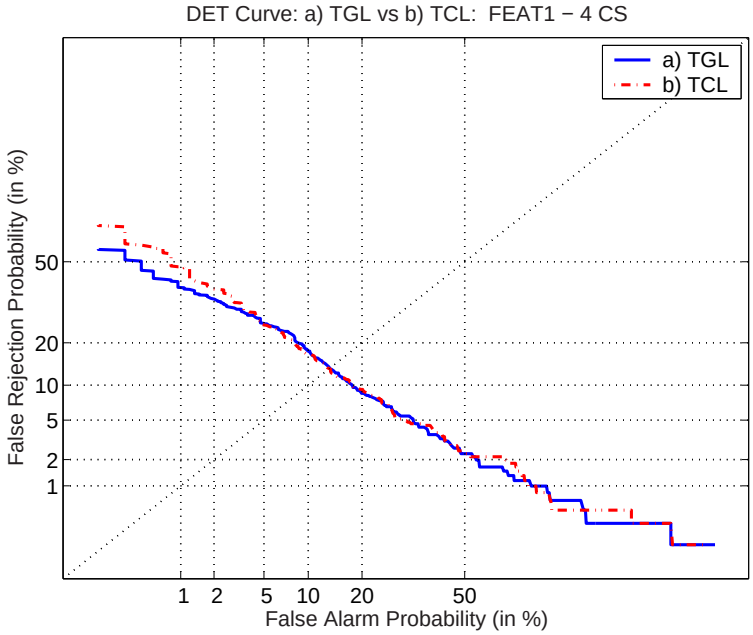


Figure 6.3: *Comparison of DET curves of experiments that trained genuine labels (TGL) and experiments that trained with classified labels (TCL) using 4 conversation sides and feature set FEAT1.*

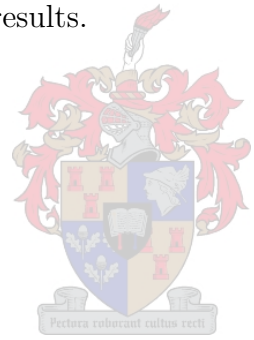
to be estimated well. The fact that more data is available for each phoneme label than is the case with words could explain why modelling substitution errors of the phoneme recogniser helps improve phonetic speaker recognition, while modelling the substitution errors of words does not improve lexical speaker recognition in this case.

6.7 Summary

This chapter deals with the use of word labels for speaker recognition. In the same way that phoneme labels generated by a phoneme spotter were used in the previous chapters for phonetic speaker verification, the words generated by a word spotter can be used for lexical speaker verification. The English language consists of a prodigious number of words, and although only a relatively small group of these words appear in Switchboard I and II, the number of unique words appearing in these datasets are still ample compared to the phoneme labels used for the phonetic speaker recognition. Subsequently, a simple approach is taken by counting single word frequencies of a selection of words. The selection of words one chooses as feature set plays an important role in the lexical approach to speaker recognition. The words appearing in the target set of a target speaker, the test

sequence and the set of words used by impostor speakers can all differ from one another. It is therefore important to include words in the feature set that are most likely to appear in all of the target, test and impostor sets. We experiment with different word selections, such as ones based on minimum word count (varying the minimum word count), speaker entropy and deviations of the log-probabilities of the UBM and target model. Experiments involving the latter two lack sufficient training data for the complexity of the problem, and result in poor accuracies. Experiments are done training classified word labels as well as training genuine word labels. The latter is time-consuming and does not perform remarkably better than training classified word labels. The best results are attained by selecting words based on a minimum word count, specifically using a minimum word count of 20 in the entire Switchboard I set.

The next chapter covers the theory involving classifier fusion. The output of the T-Norm scores of various classifiers can be combined with the aim of improving the performance of the individual classifiers. This is achieved by combining non-acoustic results (word and phoneme labels) with acoustic results.



Chapter 7

Combining Verifiers

7.1 Introduction

In the previous chapters, we dealt with the use of phoneme labels and word labels as feature sets for speaker recognition. This approach is acoustic and focuses on higher-level information of speech. The acoustic approach involves the movement of speech through the vocal tract. The most customary way of doing acoustic speaker recognition is to convert raw speech to MFCCs. Although the acoustic approach gives far better accuracies than the non-acoustic one, the latter might carry extra information, valuable to speaker recognition.

Verification may be more accurate when generated by a combination of verifiers than when generated by any of the constituent verifiers. Verifier combination can be categorised into two types: verifier selection and verifier fusion [29, 28]. Generally verifier *selection* is used when there are a few verifiers that verify elements in the same feature vector, and when the assumption is made that each verifier is "an expert" in some local area of the feature space. One verifier can be nominated to make the decision, or if there are more than one "local expert", these can be combined to make the decision. With verifier *fusion*, the verifier outputs are made comparable by scaling them over the same interval. These outputs can be fused either by averaging them or by multiplying them. In [53] it shows that, when posterior probabilities are well-estimated, the mean-combination rule and product-combination perform the same in the case of a two-class problem. Another way of fusing the verifiers is to consider the outputs as the inputs to another second-level verifier and to use pattern recognition techniques for the second-level [56].

In this chapter we compare a few fusion techniques to their constituent verifiers. The aim here is to show that the non-acoustic systems can contribute to the acoustic system. A subset of the NIST 2002 evaluation set (Switchboard II) is used in this chapter. The next chapter deals with the NIST 2004 evaluation.

7.2 The verifier scores used for combination

The following verifiers are combined:

Verifier A: The first type of T-Norm verifier uses 40 phoneme labels as feature set, and models the speakers with an HMM. (As has been done in Chapters 3 to 5). A UBM is trained and used to Dirichlet smooth the target speakers, using a prior factor equal to the number of destination links in the speaker HMM [1]. A cutoff value of 5 times the number of links is used (See Section 5.3.1 for more information about the cutoff).

Verifier B: The second type of T-Norm verifier uses the selection of word labels in Switchboard I with a minimum word count of 20. The words each speaker utters are classified, counted and normalised to obtain word probabilities. (As has been done in Chapter 6). A UBM is trained by counting each of the words used by the UBM speakers. A small value (1/(the number of words in the feature set)) is added to the word counts and these are normalised to obtain word probabilities. Each target set is trained by smoothing with the UBM, and by using a prior factor of 0.1 times the number of words in the feature set (4087 words).

Verifier C: During 2002, the *Massachusetts Institute of Technology* (MIT) provided Stellenbosch University with a set of verifier scores for each jackknife block in Switchboard II, using the target speakers and test sequences associated with each jackknife block . The acoustic system (dynamic approach) that was used, was a standard GMM-UBM system using acoustic features with a mixture UBM. The MIT verifier scores are normalised using a set of impostor scores to make it comparable to the output scores of the other two verifiers, which are T-Norm verifiers.

[1] A significance test shows that using the UBM as prior model for first-order target speakers is not a significant improvement over using the first-order models without smoothing (see Section A.8.1). Since this is not an improvement of the first-order results, it is recommended that the UBM is not used as prior model. It suggests that there is sufficient training data for first-order speaker modelling. In Chapter 8 the first-order target models are trained without using a UBM prior model.

The output scores of all three verifiers typically range from -4 to 8.

7.3 UBM, Target, Impostor and Validation Sets

Target Sets: Target speakers are trained for jackknife set 0 to 3 (jack_0 to jack_3) in Switchboard II.

Validation Set: Jack_4 (Switchboard II) is used as a validation set. Any decisions concerning the fusion of scores of the target sets are made by studying the distribution of **target** and **impostor scores**. **Target scores** are verifier scores generated from a test sequence of the target speaker being verified. **Impostor scores** are scores generated from a test sequence that is not uttered by the target speaker. The impostor and target scores from the validation set are also used for second-level training when the verifiers of the target sets are fused.

UBM Speakers: All the speakers from jack_5 to jack_9 (Switchboard II) are used as UBM speakers. The training data available for all these speakers are pooled to train a UBM.

Impostor Speakers: For Verifier *A* (phoneme-based), the impostor speakers consist of a subset of 40 speakers from Switchboard I, trained using 16 conversation sides. (The set of impostors stays the same for all the evaluations of jack_0 to jack_3, and for the evaluation of the validation set of jack_4.)

For Verifiers *B* (word-based) and *C* (MIT), the impostor speakers are selected from the remaining jackknife blocks (Switchboard II) that have not been included in the set of UBM speakers, the validation set and the target speakers of the jackknife block in question. For Verifier *B*, the impostor speakers are selected from speaker models trained using 16 conversation sides. If for instance, we use target speakers and test trials from jack_0, a validation set from jack_4, and UBM speakers from jack_5 to jack_9, the impostor speakers will consist of a subset of speakers from jack_1, jack_2 and jack_3. The same set of impostor models is used for both the target and the validation sets. The MIT scores are normalised with the associated set of impostor scores by making use of Equation 2.6. Using this setup, there are 4 sets of T-Norm scores for the validation set associated with each target set of jack_0 to jack_3 and its corresponding impostor set.

7.4 Combining the verifier output scores

7.4.1 Verifier Selection and Verifier averaging

As described in Section 7.1, classifier selection is a classifier combination technique that selects the classifier that is an "expert" in a local region of the feature space. Since word labels, phoneme labels and cepstral-based features have different feature spaces, the verifier selection technique has to be altered slightly. The MIT verifier is selected in regions where it is certain that the test sequence is uttered by either an impostor speaker or a target speaker (either relatively low or relatively high likelihood scores). In the uncertain regions, the other verifiers are incorporated in our decision by again making use of selection, or fusion by taking a weighted average of the verifier scores.

After plotting the verifier score distributions (both target and impostor distributions) of the validation set, the following was decided (inspection was done visually):

- All the verifiers seems to make few false rejection errors when the verifier scores are above 3. When the verifier score is above 3, this is referred to an area where the verifier is certain that the speaker of the test segment is a target speaker.
- All verifiers seems to make few false acceptance errors when the verifier scores are below -1. When the verifier score is below -1 this is referred to as an area where the verifier is certain that the speaker of the test segment is an impostor speaker.
- If the verifier score lies between -1 and 3, this area is referred to as an uncertain area.

The following verifier combinations are used (typical scores of all types of verifiers range from -4 to 8):

Selection of Verifier C (Acoustic MIT system)

- Verifier C is nominated as our "expert" verifier, because of its high accuracy (EERs range from typically 4% for speakers trained using 4 conversation sides to typically 2% for speakers trained using 16 conversation sides).
- This verifier is selected when it is certain that the speaker of the test segment is an impostor speaker (score $<$ -1).

- This verifier is also selected if it is certain that the speaker of the test segment is a target speaker (score > 3).
- In the cases where Verifier C is uncertain of the speaker being either a target or an impostor speaker (score lies between -1 and 3), the combination rules are followed by incorporating Verifiers A and B.

Incorporation of Verifiers A and B (Non-acoustic Systems)

- If Verifier A or B is certain of a target speaker, without the other verifier being as certain of an impostor speaker, the verifier score of the verifier that is certain is selected. The same rule is applied if Verifier A or B is certain of an impostor speaker, without the other verifier being as certain of a target speaker.
- In cases where:
  - Neither Verifier A or B is certain (both are uncertain) of a target or impostor speaker or
  - Verifier A is certain of a target speaker and verifier B is certain of an impostor speaker or vice versa

  the verifier scores are fused by weighted averaging:

$$s = \frac{s_A + s_B + 3 * s_C}{5}$$

  where s is the new fused score, and s_A, s_B and s_C are respectively scores from Verifiers A, B and C, with Verifier C granted the priority by giving its score a bigger weighting factor than the other two verifier scores.

7.4.2 Treating the output scores as inputs to a second-level verification problem

The validation set is used to train and evaluate data, and likelihood scores for the validation set are generated. These scores are grouped into target scores (scores that are obtained from target trials) and impostor scores (scores that are obtained from impostor trials).

In general, if N systems need to be fused, then two N-dimensional vectors of scores are generated: one containing the target scores of the N systems and one containing the

impostor scores of the N systems. For the experiments of this chapter $N = 3$. A target model (Model_T) is trained using the vector of target scores and an impostor model is trained (Model_I) using the vector of impostor scores.

The scores from the target set are fused:
$VTS(k)$ is the N-component vector of scores of the N systems from the k-th trial of the target set. The fusion scores are log-likelihood scores and are obtained by:

$$Score_{fusion}(n) = Score_T(n) - Score_I(n) \tag{7.1}$$

where $Score_T(k)$ is the log-likelihood score of $VTS(k)$ fitted on Model_T, and $Score_I$ is the log-likelihood score of $VTS(k)$ fitted on Model_I.

The models used in this chapter are Gaussian and GMMs (see Section 2.2.1). The GMMs used are 8 mixture GMMs and are initialised using a splitting procedure. The first iteration of the split uses a single arbitrary Gaussian Mixture component.

7.5 Experiments and results

Verifier scores are fused for the target sets of jack_0, jack_1, jack_2 and jack_3.

The following notations are used for the constituent verifiers:

- **MIT** - results using the normalised MIT scores of Verifier C in Section 7.4
- **Phonemes** - results using phoneme labels as feature set(Verifier A).
- **Words** - results using word labels as feature set(Verifiers B).

The following notations are used for the combined verifier results:

- **NA-baseline** is a fusion of the non-acoustic Verifiers A and B. A GMM model is used for the fusion.
- **S&F** - a combination of verifier selection and fusion (by weighted averaging) as described in Section 7.4.1.

- Other methods of combining verifiers are to treat the scores as the input to another verification problem and to use statistical pattern recognition methods to solve it as in Section 7.4.2. The scores are verified using a Gaussian or GMM model for the target and impostor score distributions. The results using these methods are marked **Fuse_GMM** for methods using GMM models and **Fuse_Gaussian** for methods using Gaussian models.

Figures 7.1 are the DET curves of the combined fused results of jackknife sets 0 to 3 and their constituent verifier scores, using 16 conversation sides for training. (Similar results are obtained when using 4 and 8 conversation sides for training and can be seen in Figures C.1 to C.2). Shown on the DET plots of Figure 7.1 is the ratio of false rejection rate over false alarm rate ($r = FRR/FAR$).

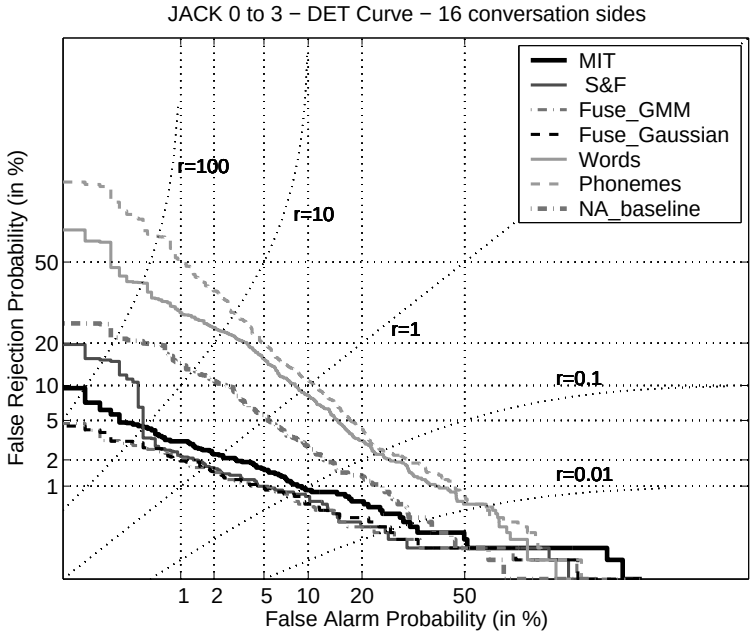


Figure 7.1: *DET curves comparing the different verifier combination techniques with their constituent verifiers, using 16 conversation sides for training.*

The fused system, NA-baseline, shows a marked improvement of the phonetic system and the lexical system . In general, Fuse_GMM and Fuse_Gaussian give the best significant improvement relative to the MIT Verifier C over the widest range of the DET curves. (The results of the significance tests can be found in Section C.2.) The results of Fuse_Gaussian and Fuse_GMM perform very similar. Since Fuse_Gaussian is a less complicated technique, it is recommended. Even though the acoustic results of MIT perform far better than the non-acoustic results (using word labels or phoneme labels), the fused results are an improvement of the acoustic results.

A summary of the EERs of these verifier combination techniques and their constituent results is given in Table 7.1. As the conversation sides increase, the given selection of word labels tends to perform relatively better than the phoneme labels as feature set.

Fusion Method	Equal Error Rate (EER)		
	4 CS	8 CS	16 CS
Verifier A (Phonemes)	13.467 %	11.222 %	10.500 %
Verifier B (Words)	14.563 %	10.779 %	9.000 %
Verifier C (MIT)	3.739 %	2.535 %	2.221 %
NA-baseline	8.950 %	6.140 %	5.359 %
Fuse_Gaussian	2.807 %	1.736 %	1.593 %
Fuse_GMM	2.829 %	1.797 %	1.641 %
S&F	2.915 %	1.970 %	1.690 %

Table 7.1: *Comparison of EERs of fusion techniques $Fuse\_Gaussian$, $Fuse\_GMM$ and $NA-baseline$ to the constituent Verifiers A, B and C, using different numbers of conversation sides (CS) for training.*

7.6 Conclusion

In this chapter we have shown that it is possible to improve the results of an acoustic verifier, by combining the acoustic verifier scores with non-acoustic verifier scores, even though the non-acoustic verifiers performed much worse than the acoustic one. Some combination techniques prove to be more useful than others. For instance, using the verifier scores as input to a second-level verifier and solving these with statistical pattern recognition analysis by training score distributions with a Gaussian model or a GMM, gives better results than the technique where verifier selection and fusion is used by weighted averaging. There is a marked difference in the results where the phonetic system (Verifier A) is fused with the lexical system (Verifier B) as compared to the results where the phonetic system and the lexical system are used on their own.

Chapter 8

NIST 2004 Evaluation

8.1 Introduction

During April 2004, the University of Stellenbosch, in collaboration with Spescom DataVoice, participated in the NIST 2004 evaluation. The 2004 evaluation used new conversational speech data collected in the Mixer Project using the Linguistic Data Consortium's new "Fishboard" platform[1]. The data is mostly conversational telephone speech in English, but includes some speech in languages other than English.

Previous evaluations have included both a limited data condition and an extended data condition. The limited data condition means that the training and test segment data for each trial consist of two minutes or less of concatenated segments of speech data, with silence intervals removed. The extended data means that the training data consists of single or multiple conversation sides, while the test data consists of single conversation sides. During the 2004 evaluation, there was no distinction between the limited and extended data conditions and also no silence removal, but rather multiple testing conditions involving the amount and type of data available for both the training and the test segments.

Section 8.2 deals with the task definition of the 2004 evaluation. Section 8.5 gives a description of the systems submitted for the evaluation, while Section 8.8 gives the results and performances of the submitted systems.

[1]See http://www.upenn.edu/mixer/

8.2 Task Definition and Evaluation Conditions

The following information is taken in parts from the NIST website[2]. The 2004 speaker recognition evaluation was limited to the broadly defined task of speaker detection, i.e. to determine whether a specified speaker is speaking during a given speech segment. Each decision is based only upon the specified test segment and the target speaker model. Use of information about other test segments and/or other target speakers is not admissible[3]. The use of the evaluation data for impostor modelling is not allowed. Each trial must be independently judged as "true" for a target trial or as "false" for an impostor trial. In addition to the detection decision, a likelihood score is required, where higher scores indicate greater confidence that the trial is a target trial.

There are 7 training segment conditions, namely 10 seconds, 30 seconds, 1 side, 3, 8 and 16 single channel conversation sides and 3 summed channel conversations. The 4 test segment conditions are 10 seconds, 30 seconds, 1 single channel conversation side and 1 summed channel conversation. A conversation side consists of approximately the last 5 minutes of a 6 minute conversation. This means that there are altogether 28 combinations of training/test conditions.

The performance of the speaker detection tests was evaluated by a detection cost function (DCF) that is defined as a weighted sum of miss and false alarm error probabilities:

$$\begin{aligned} C_{Det}(th) &= C_{FR} \times FRR(th) \times P(Target) + \\ & \quad C_{FA} \times FAR(th) \times (1 - P(Target)) \end{aligned} \tag{8.1}$$

for the threshold th, where the cost of a false rejection is $C_{FR} = 10$, the cost of a false acceptance is $C_{FA} = 1$ and the probability of a speaker being a target speaker is $P(Target) = 0.01$.

8.3 Development Data

The NIST Switchboard II corpus (the evaluation data for 2002/2003) serves as the development data for the NIST 2004 evaluation. This data consists only of **American-English**

[2] See www.itl.nist.gov/iad/894.01/tests/spk/2004/SRE-04_evalplan-v1a.pdf

[3] except as permitted for the unsupervised adaption mode condition.

Training Condition	Target Models	Test Trials	Total length of Speech
8 sides	394	16 980	±1 678 hours
16 sides	117	5064	±578 hours

Table 8.1: *A summary of the NIST 2004 evaluation data.*

speakers and **regular (landline) phone** data. All conversation sides are from the same phone number (single handset). The development data are used to select UBM speakers, impostor speakers and to choose decision thresholds.

The UBMs are trained using combined (pooled) data from jackknife sets 5 to 9 of Switchboard II. A set of 67 unique impostor speakers are selected for the T-Norm verification from jackknife blocks 1, 2 and 3. The 67 impostor models are trained using 16 conversation sides for training. The data of jackknife block 0 and 4 is used for any other validation purposes (such as choosing decision thresholds).

8.4 Evaluation Data

The NIST 2004 training and test segment data was newly collected by the *Linguistic Data Consortium* (LDC). The participating speakers took part in six-minute conversations on specified topics with people they did not know. The data includes some conversations over cellular phones.

Table 8.1 give a summary of the NIST 2004 evaluation data for training conditions of 8 and 16 single channel conversation sides and test conditions of 1 single channel conversation side.

Other than the length of the conversation sides, the evaluation data differs from the development data in that it consists of speakers other than English speakers. Most of the training data is in English, but some training conversations involving bilingual speakers are collected in Arabic, Mandarin, Russian, and Spanish. There are therefore conversations in Non-American English. The evaluation data consists of data collected using cellular phones and cordless phones, while the development data consists of data collected using landline (regular) phones.

8.5 System Description

The University of Stellenbosch, in collaboration with Spescom DataVoice, participated in the categories of 8 and 16 single channel conversation sides training conditions combined with 1 single channel conversation side test condition. We submitted 2 non-acoustic systems, SDV_2 and SDV_3, and one fused system, SDV_4, which fuses the acoustic system, SDV_0 of Spescom DataVoice, with the non-acoustic systems SDV_2 and SDV_3. SDV_2 is a lexical speaker recognition system and SDV_3 is a phonetic speaker recognition system.

8.5.1 SDV_0

SDV_0 of Spescom DataVoice is an acoustic GMM-UBM system based on MFCCs that are feature-warped according to [42].

Systems SDV_2, SDV_3 and SDV_4 were submitted by the University of Stellenbosch.

8.5.2 SDV_2

SDV_2 uses word transcriptions. The same basic approach is used as in Chapter 6. A selection of words with a minimum word count of 20 is made from Switchboard II and chosen as feature set. This feature set includes a total of 5219 word labels.

The UBM

Word transcriptions are counted. To avoid numerical problems during verification, smoothing is used: The word counts of the UBM are smoothed by adding $\frac{1}{5219}$ to all word counts. (5219 is the total number of word labels included in the feature set). The word counts are normalised to obtain word probabilities.

Training of Target and Impostor Models

The word probabilities of the UBM are used as priors, using Dirichlet estimation to smooth the target and the T-Norm impostor models. A prior factor of 0.1 times the the number of word transcriptions in the features set is used. This means that $\frac{\alpha}{\sum_i c_i} < 1/6$ (see Equation 4.6) for all the target models in jackknife set 0 of Switchboard II (excluding the models trained using 4 conversation sides).

8.5.3 SDV_3

SDV_3 uses a phonotactic approach to extract linguistic information from the training data, by using phoneme transcriptions. It uses a hidden Markov model (HMM) to model the transitions between phonemes produced by an automatic phoneme recogniser. We use the same approach as we do with the phonetic speaker system of Chapter 7, but use another phoneme recogniser model.

Automatic Phoneme Recogniser

The following information was provided by Herman Engelbrecht[4] : The phoneme recogniser consists of a context-independent phoneme spotter HMM which contains the 39 phonemes derived from the TIMIT phoneme set. The individual phonemes are each modelled with a 3-state left-to-right HMM. The phoneme models are trained on the training set of the NTIMIT speech corpus (excluding the 2 utterances spoken by each of the 630 speakers). NTIMIT consists of **American-English** speech. As input features the phoneme spotter utilises 12-dimensional MFCC features, with cepstral mean subtraction, velocity and acceleration features, and dimension reduction using *linear discriminant analysis* (LDA). LDA is a method that reduces the dimensions of the feature vector, while it maximises class separation [6]. Before the input features are extracted from the raw speech, a speech detection algorithm is used to separate the useful speech from the silences. The general concept behind the speech detection algorithm is to determine a power floor of the speech signal. All sections of speech with a frame power lower than or close to the power floor are regarded as silence and are removed.

[4]Herman Engelbrecht designed the phoneme recogniser.

The final 18-dimensional features are modelled with full-covariance Gaussian mixture models (GMMs) with 16 components. The average phoneme transcription accuracy is 50.73%[5] on the full NTIMIT testing set (excluding the 2 utterances spoken by each of the 630 speakers). Using this phoneme recogniser gives slightly better results on the development set than the phoneme recogniser used in the previous chapters.

Phonotactic Approach

The same model structure is used as described in Section 3.4.2, and the SEP of the phoneme recogniser are modelled in the PDF of the HMM. Both the UBM and the target models make use of the same state topology. The transition probabilities of the UBM are used to initialise those of the target models (no Dirichlet estimation is used). First-order results that use a UBM as prior model show no significant improvement (over the widest range of the DET curves) compared to first-order results when no smoothing is used (results are shown in Section A.8.1). This is probably due to sufficient data for first-order speaker modelling. We therefore do not use the UBM as prior model.

8.5.4 System SDV_4

System SDV_4 is a fused system of the acoustic system SDV_0 (of DataVoice) and the non-acoustic systems SDV_2 and SDV_3. For the fusion of SDV_4 the same approach is used as described in Section 7.4.2. Two separate 8 mixture GMMs are used as models to train the target and impostor scores[6]. A set of new likelihood scores is obtained after fusion by making use of Equation 7.1. As part of the evaluation, NIST required the selection of a primary (designated) system, which was compared to the primary systems of all the other participants. SDV_4 was selected as the primary system.

8.6 Choosing the Decision Threshold

NIST evaluates the data, using the cost model defined in Equation 8.2. Decision thresholds for all the systems are calculated by evaluation of the data of jackknife block 0 of

[5]See Equation 3.2 for the calculation of the phoneme error rate.

[6]It is recommended to use Gaussian models instead, since they are less complicated and produce similar results for fusion.

Switchboard II. Target models are trained and evaluated (by calculating likelihood scores) for jackknife block 0 for each of the systems SDV_2, SDV_3 and SDV_4.

The FRR and FAR of the data of jackknife 0 are calculated as a function of decision thresholds and these are used to calculate C_{Det} in Equation 8.2 for each of the systems. The decision threshold for each system is chosen as the threshold th that minimises $C_{Det}(th)$ based on the results of the **development set**. This cost is referred to as the actual DCF. When using the evaluation set, the actual DCF is not necessarily the most optimal one. The optimal DCF is the minimum value of $C_{Det}(th)$ based on the results of the **evaluation set**.

8.7 Computational Statistics

The computation time of lexical speaker models is much faster than that of phonetic speaker models. Section D.1 gives a short summary of the computational statistics of the phonetic and lexical systems[7].

8.8 Results

The NIST 2002 and 2003 evaluation used Switchboard II as their evaluation data. Switchboard II consisted of data collected in American-English, and used regular (landline) phones. Since the NIST 2004 evaluation data consists of, amongst others, non-English languages and non-American English, and uses cellular and cordless phones, our main interest is to investigate the effects of these on the overall system performances.

8.8.1 Overall Performance Including All Trials

The DET plots of Figures 8.1 and 8.2 show the overall performance of the systems. A triangle indicates the operating point at the actual DCF and a circle at the optimal DCF. These DET curves show that in general the acoustic system SDV_0 has the better performance in comparison to the fused system SDV_4.

[7]Thank you to Francois Cilliers for providing this information.

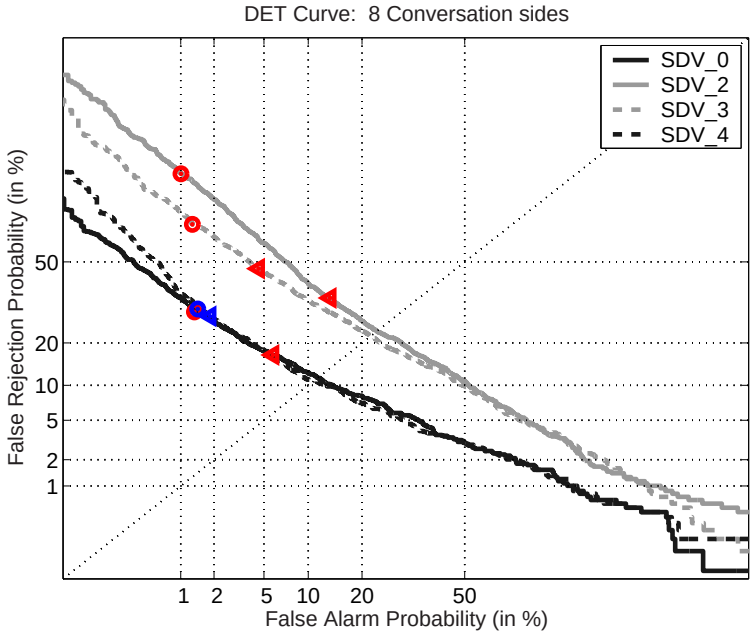


Figure 8.1: *DET curves of all the systems including all trials, using 8 (5 minute) conversation sides for training and 1 conversation side for test segment trials.*

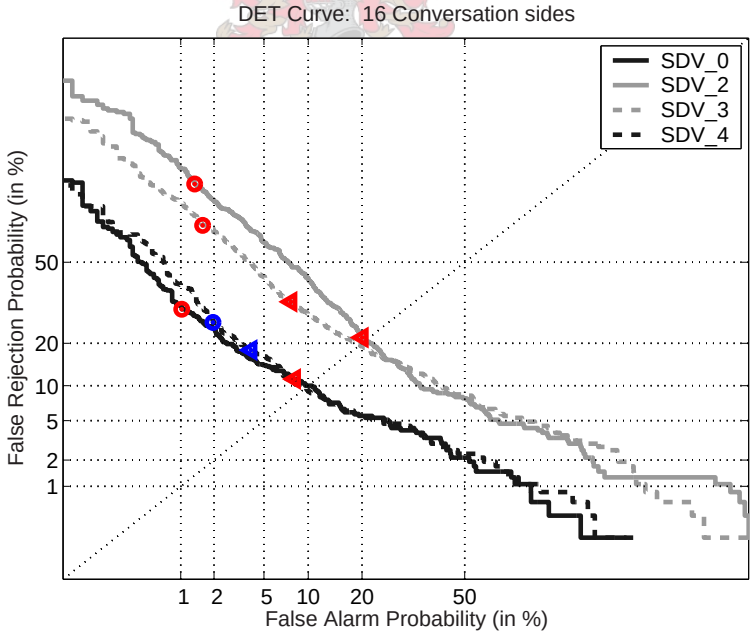


Figure 8.2: *DET curves of all the systems including all trials, using 16 (5 minute) conversation sides for training and 1 conversation side for test segment trials.*

8sides-1side			
System	Actual DCF	Ranking	Number of Primary Systems of Competing Participants
SDV-4	0.047417603307	2nd	9
SDV-0	0.076434757782	2nd	9
SDV-3	0.093589182585	3rd	9
SDV-2	0.170066073633	6th	9
16sides-1side			
System	Actual DCF	Ranking	Number of Primary Systems of Competing Participants
SDV-4	0.058770409576	3rd	5
SDV-0	0.095073183081	3rd	5
SDV-3	0.112949153999	3rd	5
SDV-2	0.221080214772	4th	5

Table 8.2: *Actual C_{DET} costs for the various systems including all trials.*

Table 8.2 shows the actual C_{Det} cost of the systems including all the trials (discarding a handful of cross-gender trials). We compare our primary system (SDV_4) as well as our sub-systems (SDV_0, SDV_2, and SDV_3) to the primary systems of the other participants. The actual C_{Det} costs are evaluated.

The system rankings are shown, which include the primary NIST systems of the competing participants as reference. (This means that if one of our systems is ranked n-th, there are $n-1$ primary systems of other participants with actual cost values that are lower than the actual cost value of our system.)

Conclusion

Even though the overall DET curve performance of SDV_4 is not the best, the fused system SDV_4 has the best (smallest) cost of all the systems, since the operating point at the actual cost is chosen relatively close to the operating point at the optimal cost, especially for training conditions of 8 conversation sides.

8.8.2 English Language Single Handset (ELSH)

English Language results are restricted to the subset where only English is used for both training and testing. The Single Handset results mean that all sides used to train a model are from the same phone number and the true target speaker also used this same handset. The performance of ELSH is similar to English Language data performance (including all handset types), since most of the English data falls in the single handset category.

Figure 8.3 shows the DET plot of ELSH for the fused system (primary system), SDV_4, using 8 conversation sides for training. The DET plots consist of DET curves for male, female and pooled gender trials. Table 8.3 shows the approximate EERs of ELSH that are estimated from the pooled gender DET plots of the various systems compared to the approximate EERs of the overall performances[8]. (The DET plots for ELSH data for the rest of the systems can be found in Figures D.1 to D.7.)

	ELSH		Overall	
System	8sides-1side	16sides-1side	8sides-1side	16sides-1side
SDV_0	11%	7%	11%	10%
SDV_2	19%	15%	24%	21%
SDV_3	20%	15%	22%	20%
SDV_4	8%	5%	11%	10%

Table 8.3: *Approximate EERs of English Language Single Handset data.*

[8]For this experiment and the experiments to follow, the DET plots of the primary system, SDV_4, using 8 conversation sides for training will be shown. Since the EERs are an adequate measurement of system performances, the EERs of all the systems will be summarised in a table. The DET plots for the rest of the systems will be shown in the appendix, since they take up so much space.

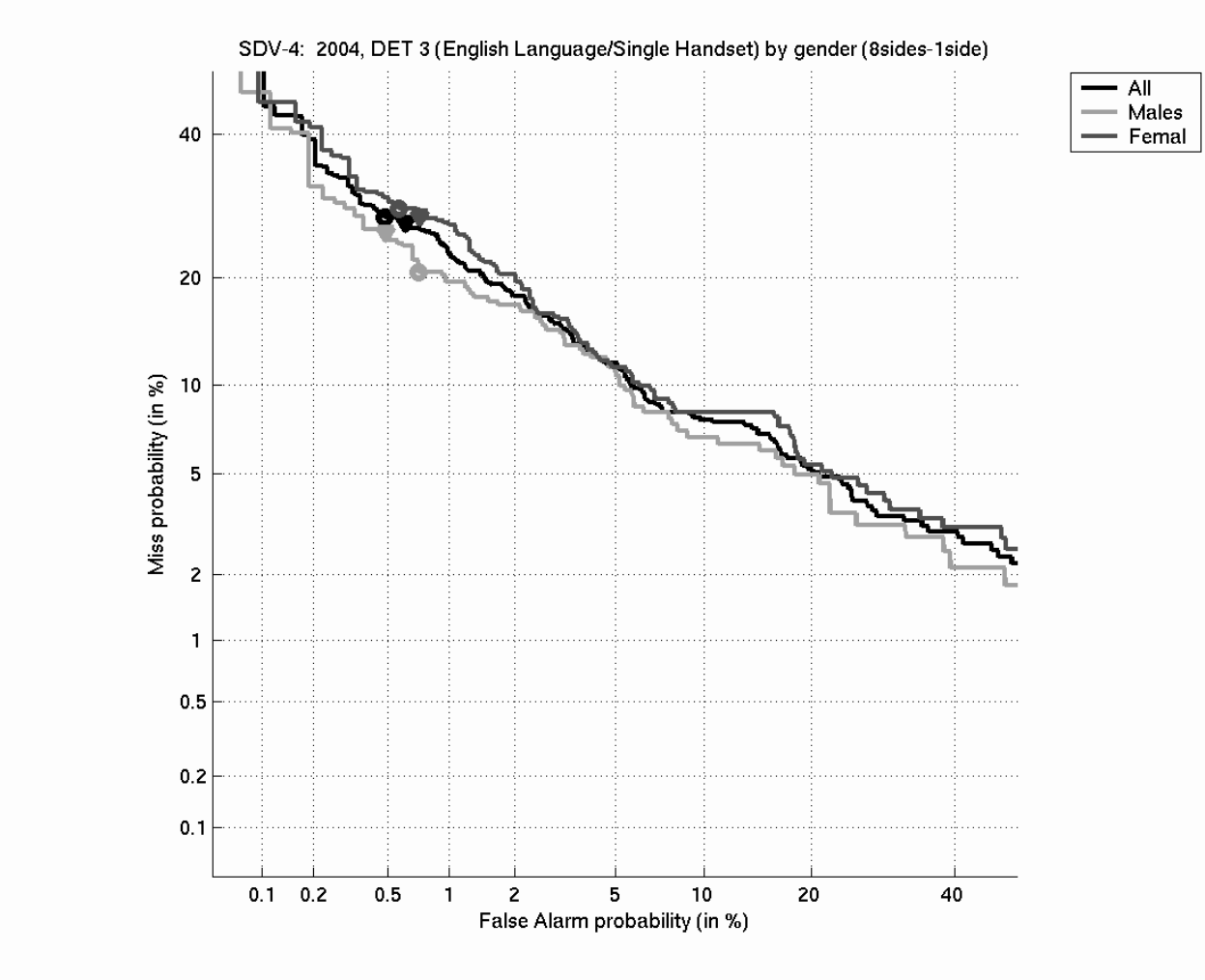


Figure 8.3: *DET curves (pooled gender, male and female trials) of SDV_4 (ELSH data), using 8 (5 minute) conversation sides for training and 1 conversation side for test segment trials.*

8sides-1side			
System	Actual DCF	Ranking	Number of Primary Systems of Competing Participants
SDV-4	0.032919756432	1st	9
SDV-0	0.071984284956	3rd	9
SDV-3	0.080246258908	4th	9
SDV-2	0.117295951152	7th	9
16sides-1side			
System	Actual DCF	Ranking	Number of Primary Systems of Competing Participants
SDV-4	0.031450342900	1st	5
SDV-3	0.085104394208	3rd	5
SDV-0	0.106231140462	3rd	5
SDV-2	0.134509525019	4th	5

Table 8.4: *Actual C_{DET} costs for the various systems for ELSH data.*

The actual DCF values for the various systems for the English Language Single Handset data are shown in Table 8.4. The results were evaluated based on the DCF values. The system rankings are also shown, which include all the primary NIST systems of the competing participants as reference. (As was done in Table 8.2.)

Conclusion

The following can be observed:

- Observing the results of the EERs and the DET curves, one can see that the fused system SDV_4 is an improvement of SDV_0. The EERs (pooled gender) using 8 conversation sides for training are approximately 11% for SDV_0 and approximately 8% for SDV_4. The EERs using 16 conversation sides for training are approximately 7% for SDV_0 and approximately 5% for SDV_4.

- The non-acoustic systems have a similar performance when compared to each other, with EERs near 20% using 8 conversation sides for training, and approximately 15% using 16 conversation sides for training.

- Non-English languages have a negative effect on the performances of especially the non-acoustic systems SDV_2 and SDV_3. This can be seen in the increase of the EERs comparing the overall system performances to ELSH.

- The actual C_{DET} costs of system SDV_4 perform the best and the operating points at these actual costs are chosen particularly close to the operating points at the optimal costs, especially using 8 conversation sides for training. The primary system SDV_4 was ranked first, using both 8 and 16 sides for training! The costs of the ELSH systems in Table 8.4 decrease (and rankings improve) compared to the costs including all the data in Table 8.2.

8.8.3 Same and Different Language Target Trials

Same language target trials are target trials where the language of the training conversations and the test conversation are the same. With different language target trials, the language of the training conversations and test conversations differ. We look at how the same and different language target trials differ in behaviour. For these results, non-target trials are fixed to include all trials where the model is trained on a single language.

Figure 8.4 shows the DET plot for same and different language target trials of the primary, fused system SDV_4, using 8 conversation sides for training. The DET plots show the performance of the target trials when the model language and the test segment language match and when they are different. Approximate EERs of the various systems are shown in Table 8.5. (The DET plots of the rest of the systems can be found in Figures D.8 to D.10 for models trained using 8 conversation sides and in Figures D.14 to D.17 for models trained using 16 conversation sides.)

Conclusion

The following can be observed:

- The fused system SDV_4 is an improvement of the acoustic system SDV_0, using same language target trials.

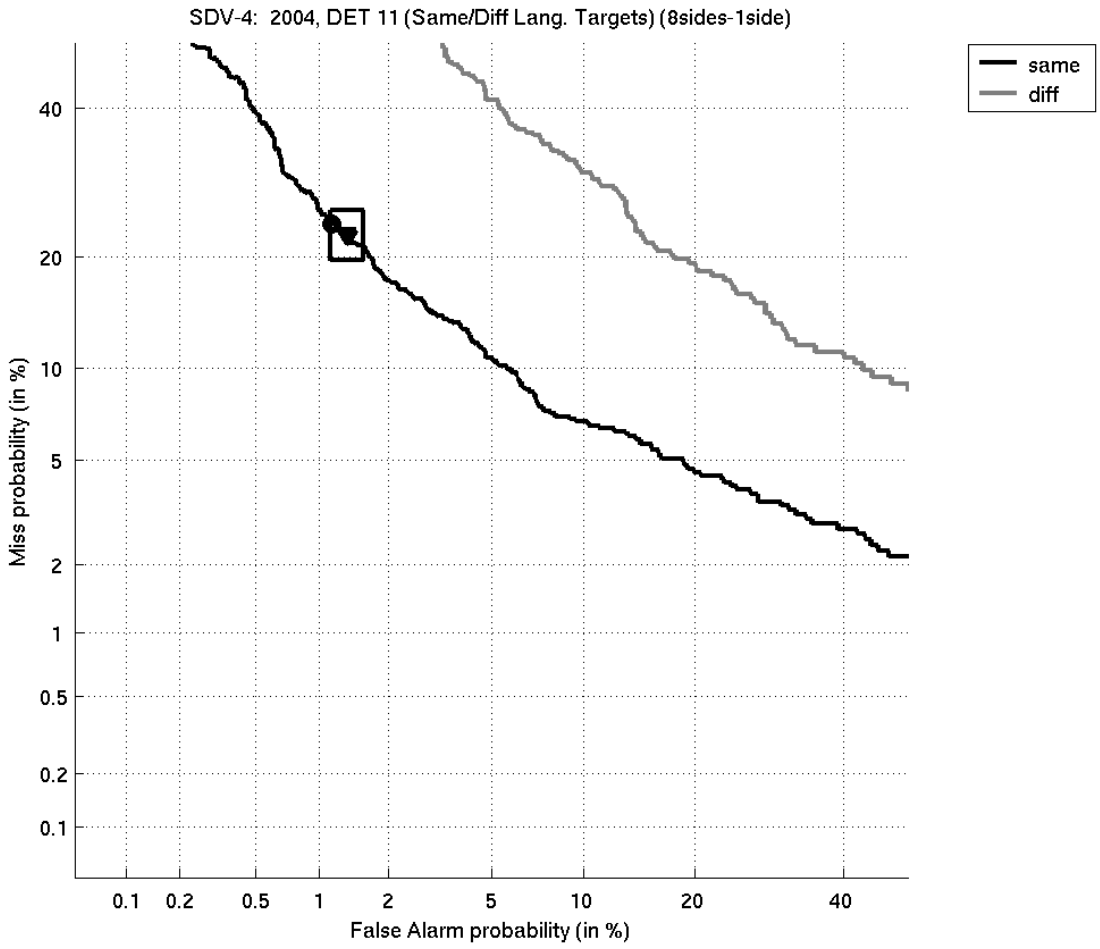


Figure 8.4: *DET curves of same and different language target trials of SDV_4, using 8 (5 minute) conversation sides for training and 1 conversation side for test segment trials.*

- The lexical system SDV_2 fares almost the same as the phonetic system SDV_3 in the same language target trials. However, the lexical system SDV_2 performs relatively poor compared to the phonetic system SDV_3 in the different language target trials.

  Although the phoneme labels are detected by a phoneme recogniser trained on American-English phonemes, quite a few of these phonemes also occur in Non-English languages. The phoneme recogniser can therefore be used with some success on these languages.

  There are not so many words in other languages that sound like English. System SDV_2 includes only English words in its feature set. It is therefore understandable

System	Target Trials				Overall	
	8sides-1side		16sides-1side		8sides-1side	16sides-1side
	Same	Different	Same	Different		
SDV_0	10%	19%	7%	20%	11%	10%
SDV_2	19%	40%	17%	> 40%	24%	21%
SDV_3	19%	35%	17%	40%	22%	20%
SDV_4	8%	20%	5%	35%	11%	10%

Table 8.5: *Approximate EERs of Same/Different Language Target trials.*

that system SDV_2 will fail with speaker recognition in cases where the test segment language differs from the model language.

- There is a marked negative influence of different language target trials on the overall performance of all the systems. EERs of same language target trials and different language target trials differ as much as 30% in SDV_4 (using 16 conversation sides for training).

8.8.4 Same and Different Language Non-Target Trials

Same language non-target trials are non-target trials where the language of the training conversations and the test conversation are the same. With different language non-target trials, the language of the training conversations and the test conversations differ. Target trials are fixed to include all trials where the model is trained on a single language. We look at how the same and different non-target trials differ in behaviour.

Figure 8.5 shows the DET plots of same and different language non-target trials of the fused system SDV_4, using 8 conversation sides for training. The DET plots show the performance of the non-target trials when the model language and the test segment language match and when they are different. The operating points at the actual and optimal costs are out of boundaries with the different language target trials and are therefore not visible on the DET curve. Table 8.6 shows the approximate EERs of the same and different language non-target trials. (The DET plots of the rest of the systems can be found in

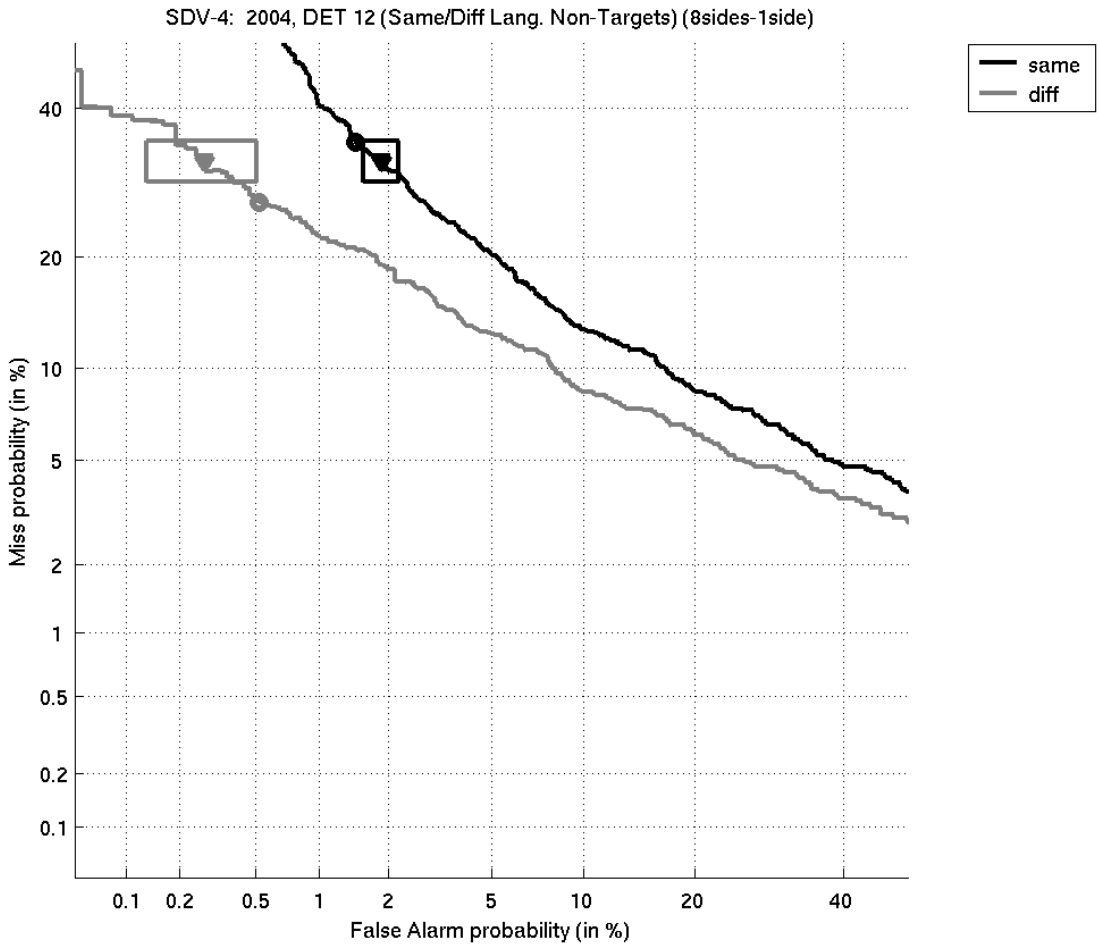


Figure 8.5: *DET curves of same and different language non-target trials of SDV_4, using 8 (5 minute) conversation sides for training and 1 conversation side for test segment trials.*

Figures D.11 to D.13 for models trained using 8 conversation sides, and in Figures D.18 to D.21 for models trained using 16 conversation sides.)

Conclusion

The following can be observed:

- In general, the same language non-target trials perform worse than the different language non-target trials. (The systems are more likely to reject speakers in different language trials.) The greatest relative difference between the same language and different language non-target trials can be seen in the lexical system SDV_2.

System	Non-Target Trials				Overall	
	8sides-1side		16sides-1side		8sides-1side	16sides-1side
	Same	Different	Same	Different		
SDV_0	14%	11%	9%	7%	11%	10%
SDV_2	27%	20%	21%	17%	24%	21%
SDV_3	25%	20%	20%	20%	22%	20%
SDV_4	14%	9%	10%	7%	11%	10%

Table 8.6: *Approximate EERs of Same/Different Language Non-target trials.*

- The lexical system SDV_2 performs worse than the phonetic system SDV_3 in the same language non-target trials, while their performance is very close in different language non-target trials. The lexical system SDV_2 struggles with the same language non-target trials in languages other than English, since the system contains only English words. As stated previously in Section 8.8.3, there are more non-English phonemes that sound like English phonemes than non-English words that sound like English words. Therefore same non-English language non-target trials are less of a problem for phonetic systems such as SDV_3.
- There is less of a negative influence of same language non-target trials than different language target trials (Table 8.5) on the overall performance of the systems.

8.8.5 The Influence of Cellular and Cordless Phones

It would be interesting to study the effect of cellular and cordless phones on system performance, where factors that have a negative influence on overall system performance, such as non-English data, are excluded. Unfortunately owing to a lack of sufficient data, this is not possible. Trials that include conversations using only regular phones in the training and the test segments are compared to the overall system performances whereby all trials (using all phone types) are included. The greatest effects of cellphone and mobile phone data can be seen in the phonetic system SDV_3 as shown in Figure 8.6. Trials using data collected solely from regular landline phones perform better than the overall performance, where cellphone and cordless phone conversations are also included.

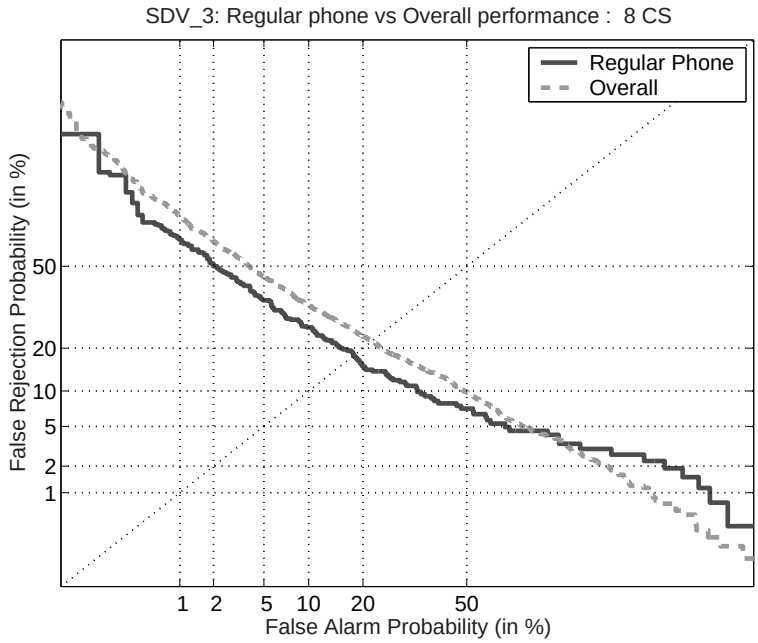


Figure 8.6: *DET curves comparing trials where only regular phones are used in comparison to the overall performance where any type of phone is used (including all trials) of SDV_3, using 8 (5 minute) conversation sides (CS) for training.*

Figure 8.7 shows three DET plots where the training data of target trials are collected from cellphone and the test segment data is collected from 1) regular landline phone 2) cellphone 3) cordless phone. All non-target trials are fixed to include all trials trained on cellphone data. These models are trained using 8 conversation sides.

Conclusion

From Figures 8.6 and 8.7 it is evident that results where the phone types of the training and test segments differ are worse than where the phone types match. (Similar results are obtained when the training data of target trials are collected from cordless phones and can be seen in Figure D.22.) This suggests that in future one should compensate for conversations not only on landline phone, but also on cellular and cordless phones. This can be done by including all phone types in the development set and the phoneme recogniser's training set.

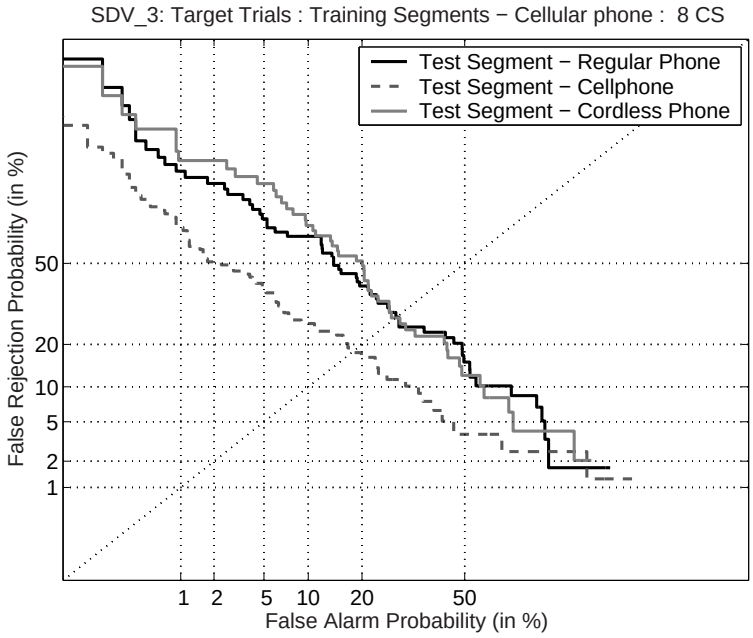


Figure 8.7: *DET curves showing the performance of trials where models are trained using 8 conversation sides and* ***cellular phones****.*

8.8.6 Comparison of Evaluation Results with Development Results

The development set contains no cellphone or cordless phone conversations and contains data collected only from American-English speakers. The T-Norm impostor speakers and UBM speakers are therefore only American-English speakers. The phoneme recogniser used is trained using speech only from American-English speakers.

The evaluation set on the other hand contains data collected not only from regular landline, but also from cellular phones and cordless phones. Most of the training conversations are in English, but these conversations include conversations collected from speakers with a dialect other than American dialect. There are also Non-English conversations involving Spanish, Russian, Arabic and Mandarin.

DET curves of results obtained using jackknife set 0 of the Development data of the various systems are shown in Figure 8.8. These results are pooled from results obtained using models trained from 8 and 16 (3 minute) conversation sides. If the results of the

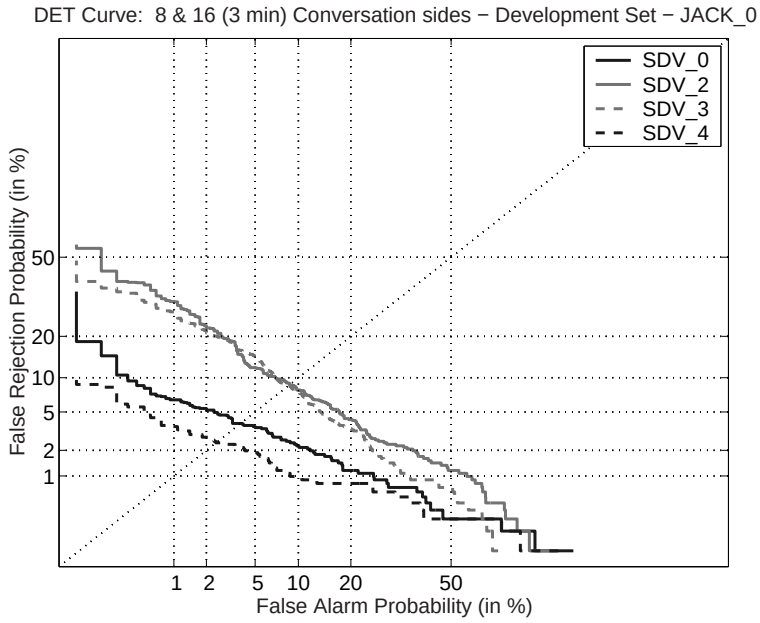


Figure 8.8: *DET curves of all the systems using the data of jackknife set 0 (jack_0) from Switchboard II. All trials of speakers trained using 8 and 16 (3 minute) conversation sides are used.*

development set in Figure 8.8 are compared to the results of systems obtained in the English Language Single Handset (Table 8.3), it can be seen that the latter fares worse, despite the conversation side length of the development data being less than that of the evaluation data. The EER of SDV_4 rises from approximately 2% to approximately 8% using 8 (5 minute) conversation sides for training and to approximately 5% using 16 (5 minute) conversation sides for training.

The drop in performance is most likely due to the fact that the T-Norm speakers from the development set are only American-English speakers, while the speakers from the evaluation data have various dialects. In addition, the phoneme recogniser was trained on American-English data and the Non-American-English data in the evaluation set is therefore slightly mismatched to that of the American-English data in the development set.

8.9 Conclusion

During the evaluation of 2004, NIST introduced a new development by using data containing other languages, as well as cellphone data. The data of previous evaluations, such as the Switchboard II data, do not contain any Non-English or cellphone data.

Data containing languages other than English affects the results drastically. The rise in EER of our primary system SDV_4 is from approximately 2% in the development set (using 8 and 16 (3 minute) conversation sides for training) to approximately 10% in the evaluation set for both 8 and 16 conversation sides training conditions (including all trials). The chief negative influence on the overall system performance is from target trials where the test segment language differs from the training segment language, especially with the non-acoustic systems SDV_2 and SDV_3 and the fused system SDV_4. Conversations from cellular and cordless phones have a negative influence on the overall performance of the phonetic system SDV_3.

The following improvements of the systems can be made:

- The phonetic system can be improved by compensating for languages other than American-English. The T-Norm speakers and the UBM speaker should include English speakers with a variety of dialects.
- A language recogniser can be employed to recognise the language of the training conversations and to then use a phoneme recogniser or word recogniser trained on this language. It is also possible to use multiple phoneme or word recognisers trained on various languages and to fuse the results obtained from these recognisers by weighted averaging. This approach would improve the results where the language of the training conversations are the same as the language of the test segment.
- Little can be done to improve the non-acoustic systems in the case where the language of the training conversations and the language of the test segment differ. The higher the level of information (words being higher-level information than phonemes), the more the limit of the non-acoustic system. Therefore it is best not to fuse the acoustic system with the non-acoustic systems in cases where the language of the training conversations differs from the language of the test segment, and to use only the acoustic system.
- Compensation for phone types other than regular phones can be made by including UBM and T-Norm speakers that have conversations over cellular and cordless phones.

Chapter 9

Conclusion

9.1 Introduction

This study concerns non-acoustic, text-independent speaker recognition based on the phonetic and lexical features of speakers instead of traditional acoustic features. Where traditional acoustic speaker recognition involves physical aspects of the vocal tract, such as pitch, non-acoustic speaker recognition is based on the use of certain words, phrases or idiosyncratic pronunciation. The interest in statistical non-acoustic speaker recognition originates from the way the human listener distinguishes amongst familiar speakers. A listener uses not only the low-level information such as pitch, but also the high-level information of idiosyncrasies. In this study phonetic features and word labels are explored as feature sets for speaker recognition. The NIST Switchboard II corpus is used in most of the experiments in this study. The data is conversational data in American-English.

9.2 Phonetic Speaker Recognition

9.2.1 The Phoneme Recogniser System

Phonetic speaker recognition is done by using classified phoneme labels as feature set. The raw speech signal is digitally processed and MFCCs are used as the input of an HMM-based phoneme recogniser. The phoneme recogniser then classifies the phoneme labels.

9.2.2 Speaker Model Structure

Fully connected ergodic HMMs are used to model the time dependencies among the phoneme labels of speakers. The HMMs are discrete with one state per phoneme label. Experiments are done using first- and second-order HMMs. Modelling the substitution errors of the phoneme recogniser in the PDF of the HMMs improves the speaker recognition system substantially.

9.2.3 Smoothing of Speaker Models

Because of data scarcity, smoothing is needed especially when using second-order HMMs. To address this problem, a few smoothing possibilities are investigated and evaluated via their EERs. Merging the phoneme labels to create more data per link seems less valuable for speaker recognition than smoothing transition probabilities using MAP adaptation with Dirichlet prior probabilities. The transition probabilities of a UBM are used to smooth the transition probabilities of TSMs, and the transition probabilities of first-order TSMs are employed to smooth the transition probabilities of second-order TSMs.

Evaluation of the DET curves and the McNemar significance tests show that smoothing second-order TSMs with first-order TSMs is an improvement over second-order results without smoothing, but not an improvement over first-order results. This is attributed to insufficient data to model second-order effects. Second-order experiments with smoothing would be worth re-evaluating when more data becomes available. Neither is smoothing first-order TSMs with a UBM prior model an improvement over first-order results without smoothing. This is probably due to sufficient data for first-order speaker modelling.

9.3 Lexical Speaker Recognition

NIST provided word transcriptions for Switchboard I and II. These classified word labels are used as feature sets for speaker recognition. The effect of different word selections are studied. Words with a word count above a chosen minimum threshold (in the entire Switchboard I) are included in the feature set and the threshold is varied. The best results are obtained by choosing words with a minimum word count of 20. In addition speaker-specific word selections are studied by having different word selections as feature set for each speaker model. This is done by using the speaker entropy of words and by

evaluating the log-probabilities of words in the UBM and TSM. Because of data scarcity, these techniques give poor results.

9.4 Verifier Combination

It is possible to use non-acoustic results to improve acoustic results. Verifier combination techniques are used to combine verifier scores of an acoustic verifier with verifier scores of non-acoustic verifiers (one using phoneme labels and the other using word labels for speaker recognition). A combination of verifier selection and fusion by means of weighted averaging of the verifier scores are investigated, where the verifier output scores serve as the input to a second-level verifier. The second-level verifier can be configured by means of statistical pattern recognition methods and gives the best improvement compared to the results of the acoustic system.

9.5 NIST 2004 Evaluation

During the 2004 evaluation, NIST used new conversational speech data collected in the Mixer Project using the Linguistic Data Consortium's new *Fishboard* platform. The 2004 evaluation not only included conversational telephone speech in English, but also contained some speech in languages other than English.

The University of Stellenbosch, in collaboration with Spescom DataVoice, participated in the NIST 2004 evaluation. The University of Stellenbosch submitted two non-acoustic systems (lexical and phonetic) and a fused system which is a fusion of the acoustic system supplied by Spescom DataVoice and the two non-acoustic systems of the University of Stellenbosch. Switchboard II serves as the development set and the T-Norm speakers and UBM speakers are selected from this development set. The system of Spescom DataVoice is an acoustic GMM-UBM system based on MFCCs that are feature-warped according to [42].

The phonetic system has the same structure as the phonetic system described in Section 9.2, i.e. it uses fully-connected, ergodic HMMs with one state per phoneme label. It models the substitution probability errors of the phoneme recogniser and uses no Dirichlet smoothing of the transition probabilities [1].

[1] The phoneme recogniser is a different one from the one described in Section 9.2.1.

The lexical system uses the same structure as described in Section 9.3. Classified words are counted and normalised to obtain classified word probabilities. The word probabilities of the UBM are used to smooth those of the target models.

The following observations are made:

- The non-acoustic systems are negatively influenced by Non-English languages, especially in the case where the language of the training segments differs from the language of the test segment. The development data contain only American-English speech and the non-acoustic systems therefore do not compensate for any non-English languages or for English languages with a dialect other than American.
- The negative influence of conversations using cellular or cordless phones is evident, especially on the phonetic system.

It is therefore suggested that, where possible, a development set be used that compensates for languages other than American-English and for regular landline phones. A language recogniser can be employed to detect the language of the training and the test segments, and should they differ, it is suggested that only the verifier output of the acoustic system be used and that the output scores of the acoustic system not be fused with those of the non-acoustic systems. Where the language of the test segment and the training segments is the same, a feature set should be used that contains words or phonemes in the same language.

9.6 Recommendations

The first step when designing a phonetic speaker recognition system is the phoneme recogniser. Should the phoneme recogniser be poor, the phonetic speaker recognition system will more than likely perform poorly. It is therefore important to design a phoneme recogniser of high quality, as the phonetic speaker recognition system is prone to be sensitive to the phoneme recogniser. In this study the different configurations of the phoneme recogniser system are not explored, but a possible way of improving it would be to use context-dependent phonemes as features. Another way is to explore the number of GMM mixtures used for the phoneme recogniser. Modelling of other phoneme recogniser errors, such as deletion and insertion errors, should be examined, since modelling the substitution errors of the phoneme recogniser improves the phonetic system a great deal.

Word frequencies are counted in the lexical approach to speaker recognition, but the use of word N-grams should also be investigated.

As mentioned before, NIST is progressing towards Non-English languages and towards cellular and cordless phone conversations. It is advisable that a language recogniser be employed to recognise the language of the training and the test segments. Feature sets can then be used in the recognised language. Subsequently, fusion of the acoustic system with the non-acoustic systems should be used only in cases where the language of all or most of the training segments is the same as the language of the test segment. The phoneme recogniser and the word recogniser should also include dialects other than American-English and should also contain conversations over cellular and cordless phones.

9.7 Final Conclusion

The best results based on phonetic speaker recognition are obtained by using first-order HMMs. Owing to data scarcity, the use of higher-order HMMs is not recommended, should the data of Switchboard II be used. Using prior models to Dirichlet smooth the first-order and second-order target models shows no significant improvement over the first-order results without smoothing. Word frequencies are counted in the lexical approach to speaker recognition.

Non-acoustic speaker recognition systems such as a lexical system and a phonetic system prove to be useful when combined with an acoustic system. The fused system can produce better results than the acoustic system. Where the test segment language differs from the language of the training segment, non-acoustic systems, which use higher-level information than an acoustic system, are more sensitive to these language differences than the acoustic system.

Bibliography

[1] ANDREWS, W. D., KOHLER, M. A., and CAMPBELL, J. P., "Phonetic speaker recognition." Proc. Eurospeech, 2001.

[2] ANDREWS, W. D., KOHLER, M. A., CAMPBELL, J. P., GODFREY, J., and HERNANDEZ-CORDERO, J., "Gender-dependent phonetic refraction for speaker recognition." Proc. IEEE ICASSP, 2002.

[3] ASSALEH, T. and MAMMONE, R., "Robust Cepstral Features For Speaker Identification." *Proc. IEEE ICASSP*, 1994, pp. I.129–I.132.

[4] AUCKENTHALER, R. *et al.*, "Score Normalization for Text-Independent Speaker Verification Systems." *Digital Signal Processing*, 2000, Vol. 10, pp. 42–54.

[5] BENNANI, Y. and GALLINARI, P., "On the use of TDNN-extracted features information in talker identification." *Proc. IEEE ICASSP*, 1991, pp. 385–388.

[6] BISHOP, C., *Neural Networks for Pattern Recognition*. Oxford: Clarendon Press, 1995.

[7] CAREY, M., PARRIS, E., and BRIDLE, J., "A Speaker Verification System Using Alpha-Nets." *Proc. IEEE ICASSP*, 1991, pp. 397–400.

[8] CASTELLANO, P., SLOMKA, S., and SRIDHARAN, S., "Telephone Based Speaker Recognition Using multiple Binary Classifier And Gaussian Mixture Models." *Proc. IEEE ICASSP*, 1997, pp. 1075–1078.

[9] CERNOCKY, J., PETROVSKA-DELACRTAZ, D., PIGEON, S., VERLINDE, P., and CHOLLET, G., "A segmental approach to text-independent speaker verification." http://www.citeseer.nj.nec.com/234693.html.

[10] DELLER, J., PROAKIS, J., and HANSEN, J., *Discrete-Time Processing Of Speech Signals*. Mcmillan Publishing Company, 1993.

[11] DODDINGTON, D., "Speaker recognition based on idiolectal differences between speakers." Proc. Eurospeech, January 2001.

[12] DU PREEZ, J. A., "Efficient training of high-order Hidden Markov Models using first-order representations." *Computer Speech and Language*, January 1998, Vol. 12, No. 1, pp. 23–39.

[13] ENGELBRECHT, H. A., "Dynamic key-phrase speaker verification." Report for Final Year Project, University of Stellenbosch, October 2000.

[14] FAKOTAKIS, N., GEORGILA, K., and TSOPANOGLOU, A., "A continuous HMM Text-Independent Speaker Recognition System Based on Vowel Spotting." *Proc. Eurospeech*, 1997.

[15] FARREL, K. and MAMMON, R., "Speaker Identification using Neural Tree Networks." *Proc. IEEE ICASSP*, 1994, pp. I.165–I.168.

[16] GILLICK, L. and COX, S., "Some Statistical Issues In the Comparison of Speech Recognition Algorithms." *Proc. IEEE ICASSP*, 1989, pp. 532–535.

[17] GISH *et al.*, "Text-independent speaker identification over telephone channels." *Proc. IEEE ICASSP*, 1985, pp. 379–382.

[18] GISH, H. and SCHMIDT, "Text-Independent Speaker Identification." *IEEE Signal Processing Magazine, IEEE*, October 1994, pp. 1437–62.

[19] GISH, H., SCHMIDT, M., and MIELKE, A., "A Robust, Segmental Method For Text Independent Speaker Identification." *Proc. IEEE ICASSP*, 1994, pp. I.145–I.148.

[20] GREENBERG, S., HOLLENBACK, J., and ELLIS, D., "Insights Into Spoken Language Gleaned From Phonetic Transcription Of the Switchboard Corpus." *ICSLP*, 1996.

[21] HIGGIN, A., BAHLE, L., and PORTER, J., "Voice Identification Using Nearest-Neighbor Distance Measure." *Proc. IEEE ICASSP*, 1993, pp. II.375–II.378.

[22] HIROAKI, H., "Text-independent Speaker Recognition Using Neural Networks." *Proc. IEEE ICASSP*, 1992, pp. II.153–II.156.

[23] HUANG, X., ACERO, A., and HON, H., *Spoken Language Processing: A Guide to Theory, Algortihm, and System Development*. Prentice Hall PTR, 1993.

[24] JANKOWSI, C., KALYHANSWAMY, A., BASSON, S., and SPITZ, J., "NTIMIT: A Phonetically Balanced, Continuous Speech, Telephone Bandwidth Speech Database." *Proc. IEEE ICASSP*, 1990, pp. 109–112.

[25] JIN, Q., SCHULTZ, T., and WAIBEL, A., "Phonetic speaker identification." http://www-2.cs.cmu.edu/ tanja/MyPublications.html, 2002.

[26] JIN, Q., SCHULTZ, T., and WAIBEL, A., "Speaker identification using multilingual phone strings." http://www-2.cs.cmu.edu/ tanja/MyPublications.html, 2002.

[27] KAO, Y., RAJASEKARAN, P., and BARA, J., "Free-text speaker identification over long distance telephone channel using hypothesized phonetic segmentation." *Proc. IEEE ICASSP*, 1992, Vol. 2, pp. 177–180.

[28] KUNCHEVA, L., "Switching Between Selection and Fusion in Combining Classifiers: An Experiment." *IEEE Transactions on SMC, Part B,32*, 2002, pp. 146–156.

[29] KUNCHEVA, L., BEZDEK, J., and DUIN, R., "Decision Templates for Multiple Classifier Fusion: an Experimental Comparison." *Pattern Recognition.*, 2001, Vol. 34, pp. 299–314.

[30] LAMEL, L. and GAUVAIN, J.-L., "Speaker Recognition with the Swithcboard Corpus." *Proc. IEEE ICASSP*, 1997, pp. 1067–1070.

[31] LEE, K. F., "Speaker-independent Phone Recognition using Hidden Markov models." *IEEE Trans. on Acoustics, Speech, and Signal Processing*, 1989, Vol. 37, No. 11, No. 11, pp. 1641–1648.

[32] MACKAY, J. and BAUMAN PETO, L. C., "A hierarchical Dirichlet Language Model." *Natural Language Engineering*, 1995, Vol. 1, No. 3, No. 3, pp. 1–19.

[33] MARKEL, J. and OSHIKA, A. J., B. GRAY, "Long-term feature averaging for speaker recognition." *IEEE Trans. Acoust., Speech, Signal Processing,*, August 1977, Vol. 35, No. 10, pp. 330–337.

[34] MARTIN, A., DODDINGTON, G., KAMM, T., ORDOWSKI, M., and PRZYBOCKI, M., "The DET Curve in Assessment of Detection Task Performance." in *Proc. Eurospeech '97*, (Rhodes, Greece), pp. 1895–1898, 1997.

[35] MATSUI, T. and FURUI, S., "Comparison of Text-Independent Speaker Recognition Methods Using VQ-Distortion and Discrete/Continuous HMMS." *Proc. IEEE ICASSP*, 1992, pp. II.157–II.160.

[36] MORENO, P. J., "Speech Recognition in Telephone Environments." Master's thesis, Carnegie Mellon University, Pittsburgh, Pennsylvania, December 1992.

[37] NAIK, J. and LUBENSKY, D., "A Hybrid HMM-MLP Speaker Verification Algorithm for Telephone Speech." *Proc. IEEE ICASSP*, 1994, pp. I.153–I.156.

[38] NAVRATIL, J., JIN, Q., SCHULTZ, T., and WAIBEL, A., "Phonetic Speaker Recognition Using Maximum-likelihood Binary-Decision Tree Models." *Proc. IEEE ICASSP*, 2003.

[39] NAVRATIL, J. and RAMASWAMY, G. N., "The Awe and Mystery of T-Norm." *EuroSpeech*, 2003.

[40] OGLESBY, J. and MASON, J., "Radial Basis Function Networks For Speaker Recognition." *Proc. IEEE ICASSP*, 1991, pp. 393–396.

[41] OLSEN, J., "Speaker Verification Based On Phonetic Decision Making." *Proc. Eurospeech*, 1997, pp. 1375–1378.

[42] PELECANOS, J. and SRIDHARAN, S., "Feature Warping for Robust Speaker verification." *Proceedings: A Speaker Odyssey*, 2001, pp. 213–218.

[43] PEYTON, Z. and PEEBLES, J., *Probability, Random Variables, and Random Signal Principles*. Third edition. McGraw-Hill,Inc., 1993.

[44] RABINER, L. and JUANG, B., "An Introduction to Hidden Markov Models." *IEEE ASSP Magazine*, January 1986, pp. 1622–1635.

[45] REYNOLDS, D. A. and ROSE, R. C., "Robust Text-Independent Speaker Identification Using Gaussian Mixture Speaker Models." *IEEE Signal Processing Magazine, IEEE*, January 1995, Vol. 3, No. 1, pp. 72–83.

[46] ROSENBERG, A. E., LEE, C.-H., and GOKEN, S., "Connected Word Talker Verification Using Whole Word HIdden Markov Models." *Proc. IEEE ICASSP*, 1991, pp. 381–384.

[47] RUDASI, L. and ZAHORIAN, S. A., "Text-independent talker identification with neural networks." *Proc. IEEE ICASSP*, May 1991, pp. 389–392.

[48] SAVIC, M. and SORENSEN, J., "Phoneme Based Speaker Verification." *Proc. IEEE ICASSP*, 1992, pp. II.165–II.168.

[49] SCHNEIDER, T. D., "Information theory primer." ftp://ftp.ncifcrf.gov/pub/delila/primer.ps, June 2002.

[50] SONMEZ, M., HECK, L., and WEINTRAUB, E., M. ADN SHRIBER, "A Lognormal Tied Mixtrure Model Of Pitch For Prosody-Based Speaker Recognition." *Proc. Eurospeech*, 1997, Vol. 3, pp. 1391–1394.

[51] SOONG, F. *et al.*, "A vector quantization approach to speaker recognition." *Proc. IEEE ICASSP*, 1985, pp. 387–390.

[52] TADJ, C., DUMOUCHEL, P., and GANG, Y., "N-best GMM's for Speaker Identification." *Proc. Eurospeech*, 1997, Vol. 5, pp. 2295–2298.

[53] TAX, D., VAN BREUKELEN, M., DUIN, R. P., and KITTLER, J., "Combining Multiple classifiers by averaging or by multiplying?." *Pattern Recognition.*, 2000, pp. 1475–1485.

[54] WEBB, J. and RISSANEN, E., "Speaker Identification Experiment Using HMMs." *Proc. IEEE ICASSP*, 1993, pp. II.387–II.390.

[55] WENNDT, S. and SHAMSUNDER, S., "Bispectrum Features for Robust Speaker Identification." *Proc. IEEE ICASSP*, 1997, pp. 1095–1097.

[56] WOLPERT, D., "Stacked generalization." *Neural Networks*, 1992, Vol. 5, No. 2, No. 2, pp. 241–260.

[57] YU, G. and GISH, H., "Identification of Speakers Engaged in Dialog." *Proc. IEEE ICASSP*, 1993, pp. II.383–II.386.

Appendix A

Phonetic Speaker Recognition

A.1 Modelling Phoneme Recogniser Errors

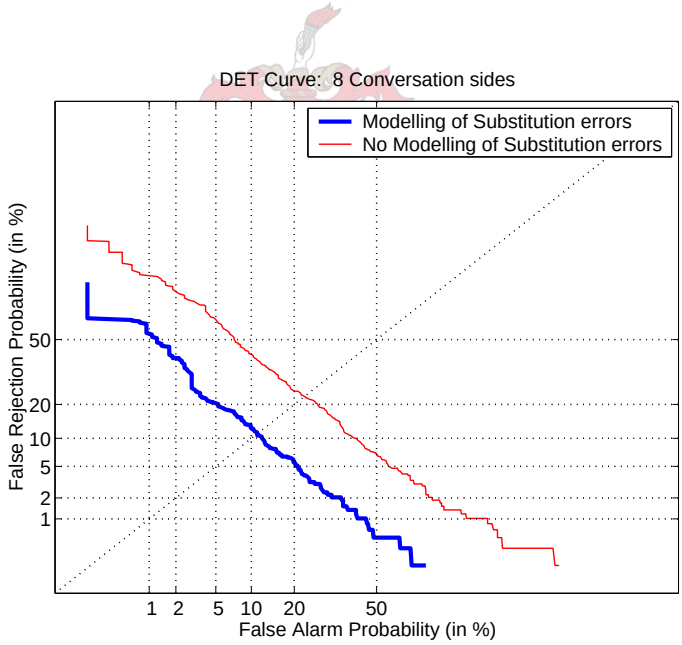


Figure A.1: *DET curves for modelling of substitution errors and no modelling thereof, using 8 conversation sides for training.*

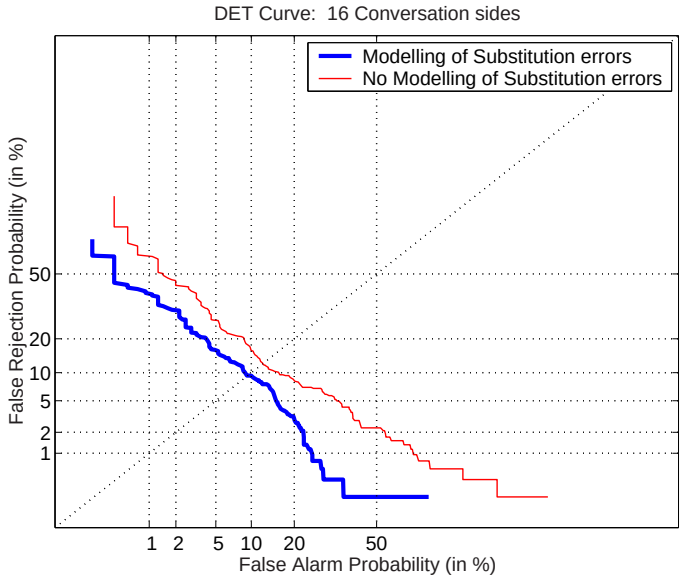


Figure A.2: *DET curves for modelling of substitution errors and no modelling thereof, using 16 conversation sides for training.*

A.2 First and Second-Order Experiments without Smoothing

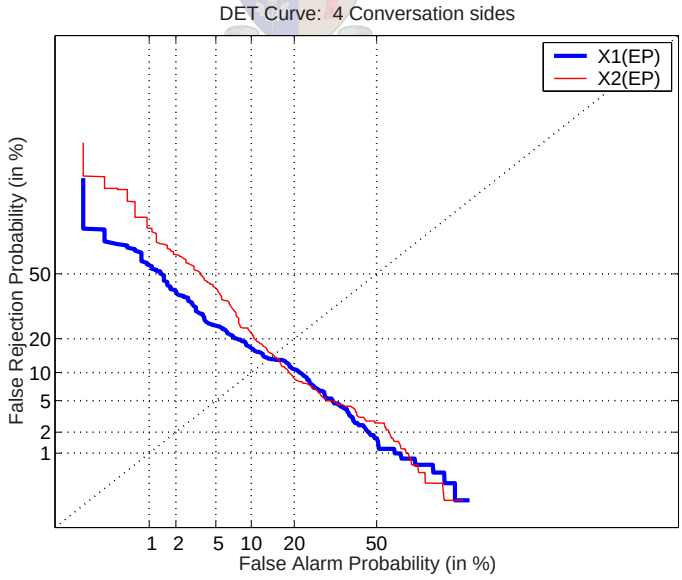


Figure A.3: *DET curves for first-order, X1(EP), and second-order experiments, X2(EP), using 4 conversation sides for training: Initialisation with equal probabilities.*

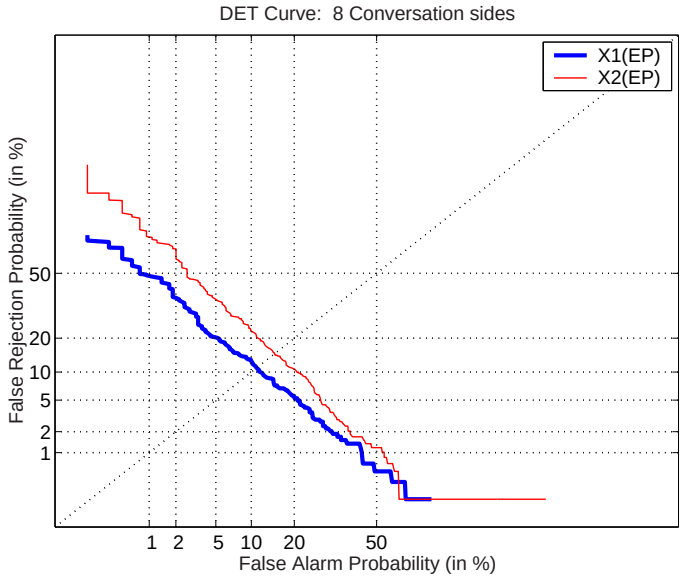


Figure A.4: *DET curves for first-order, X1(EP), and second-order experiments, X2(EP), using 8 conversation sides for training: Initialisation with equal probabilities.*

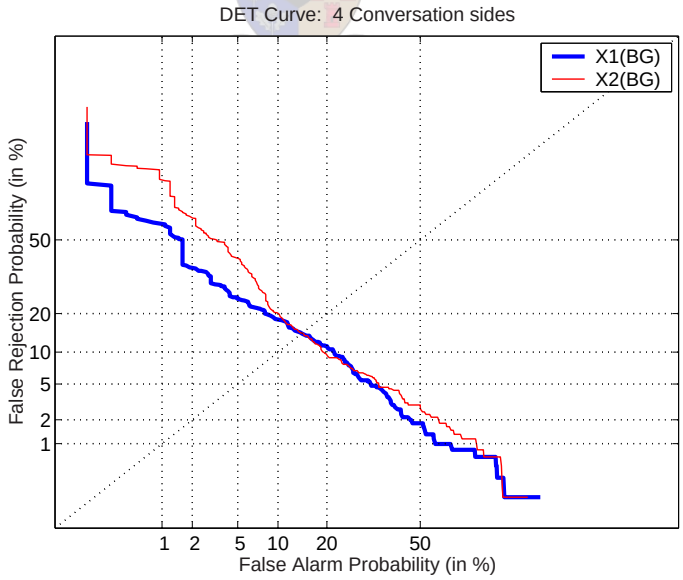


Figure A.5: *DET curves for first-order, X1(BG), and second-order experiments, X2(BG), using 4 conversation sides for training: Initialisation with normalised bigram counts.*

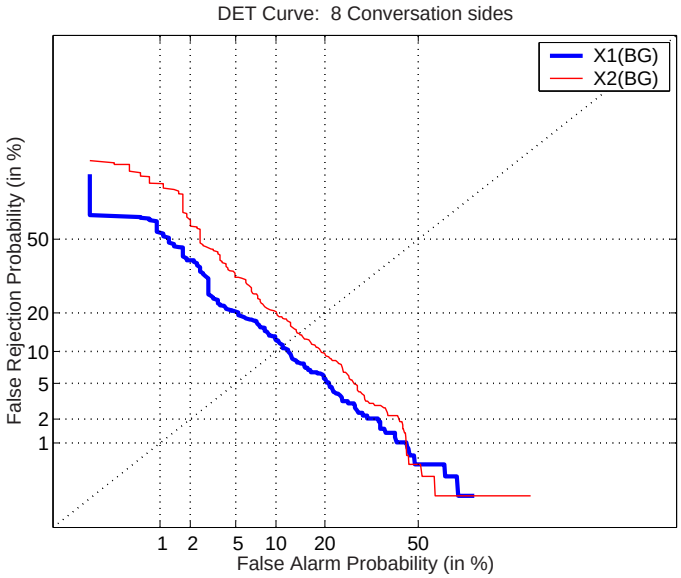


Figure A.6: *DET curves for first-order, X1(BG), and second-order experiments, X2(BG), using 8 conversation sides for training: Initialisation with normalised bigram counts.*

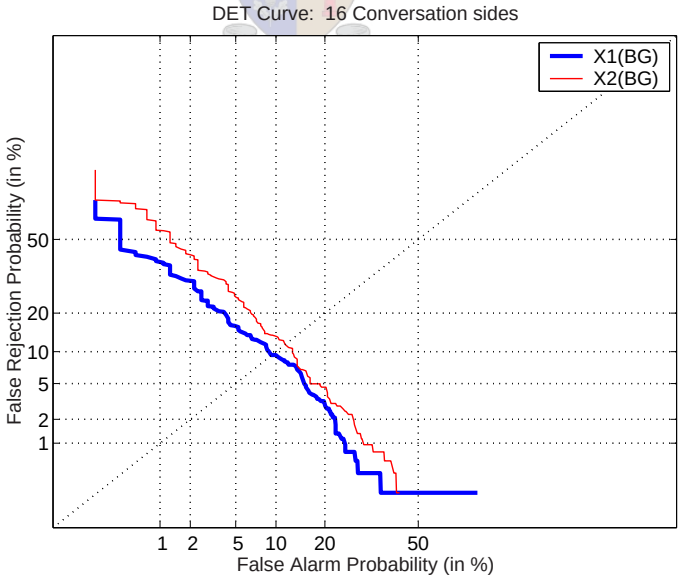


Figure A.7: *DET curves for first-order, X1(BG), and second-order experiments, X2(BG), using 16 conversation sides for training: Initialisation with normalised bigram counts.*

A.3 Merging Labels

When merging labels in order to decrease the number of links in the HMM, genuine labels are merged with only other genuine labels. (See Section 4.2.1). When labels merging is applied to a more general application, then one can also merge a common label with a classified-only label or a genuine-only label with a classified-only label.

Figure A.8 illustrates the substitution counts before label merging takes place when applied generally. $L1$ and $L2$ are the labels that are going to be merged. Genuine labels $L1$ and $L2$ are indexed by i_{L1} and i_{L2} and classified labels $L1$ and $L2$ are indexed by j_{L1} and j_{L2}. Vertical light grey areas indicate the substitution counts of a genuine label that is substituted with classified labels $L1$ or $L2$. Horizontal light grey areas indicate the substitution counts of genuine labels $L1$ or $L2$ that are substituted with a classified label. Dark grey areas indicate substitution counts where $L1$ is substituted with either $L1$ or $L2$, or $L2$ is substituted with either $L1$ or $L2$. Visually interpreted, these are the intersections of the vertical light grey regions with the horisontal light grey regions. Let L_{new} be the new merged label and L_{other} be any other label. After the new substitution counts have been calculated, the merged label L_{new} is now a common label in all four cases of the merging of labels.

For the merging of a common label $L1$ with a genuine-only label $L2$, the new substitution counts are

$$c^*_{v_{L_{new}} w_{L_{new}}} = c_{i_{L1} j_{L1}} + c_{i_{L1} j_{L2}} \tag{A.1}$$

$$c^*_{v_{L_{new}} w_{L_{other}}} = c_{i_{L1} j_{L_{other}}} + c_{j_{L2} i_{L_{other}}} \tag{A.2}$$

$$c^*_{v_{L_{other}} w_{L_{new}}} = c_{i_{L_{other}} j_{L1}} \tag{A.3}$$

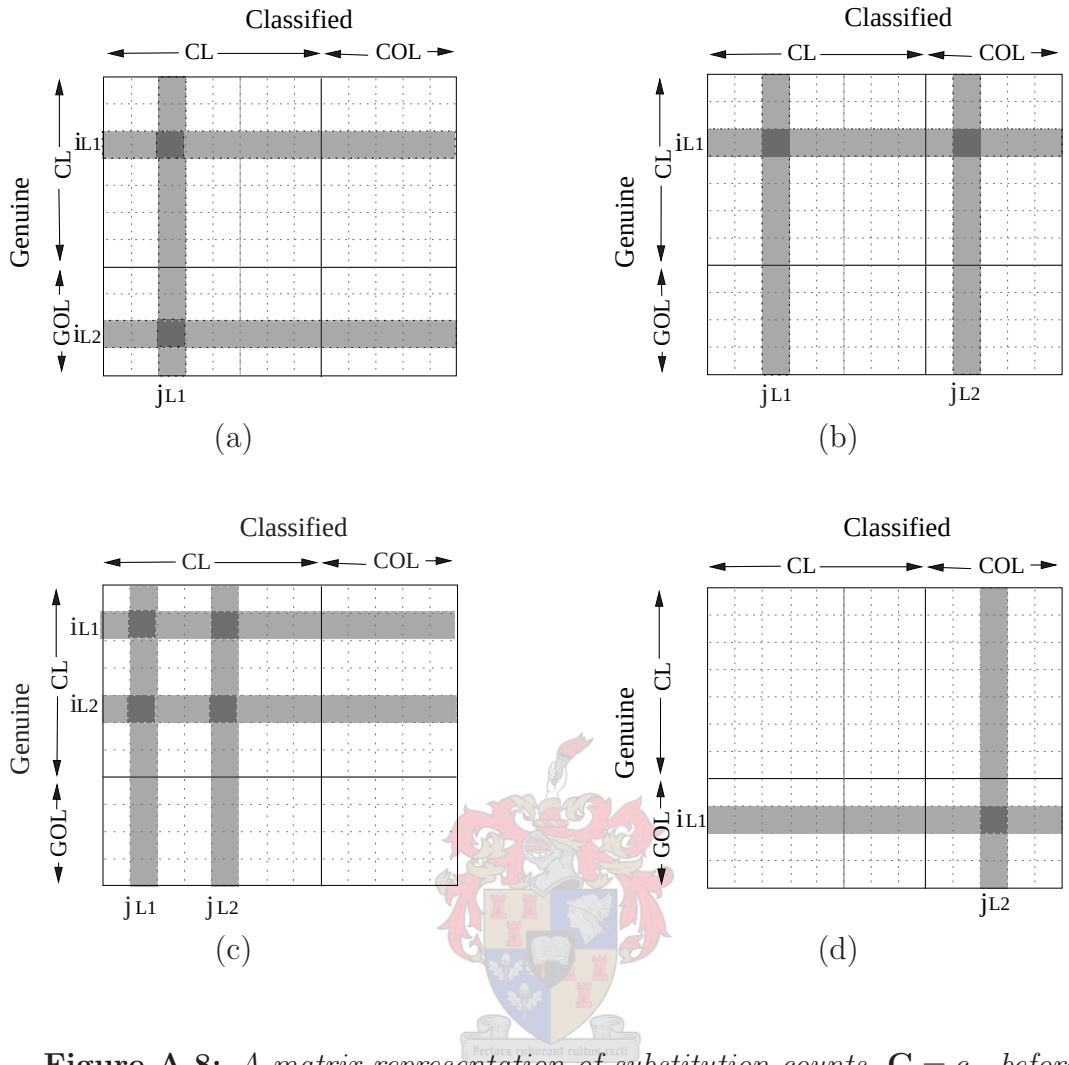


Figure A.8: *A matrix representation of substitution counts,* $\mathbf{C} = c_{ij}$ *before label merging takes place, illustrating the computation of new substitution counts for merged labels, when merging: (a) A common label with a genuine-only label and (b) A common label with a common label (c) A common label with a classified-only label and (d) A genuine-only label with a classified-only label*

For the merging of a common label $L1$ with a classified-only label $L2$, the new substitution counts are

$$c^*_{v_{L_{new}} w_{L_{new}}} = c_{i_{L1} j_{L1}} + c_{i_{L1} j_{L2}} \tag{A.4}$$

$$c^*_{v_{L_{new}} w_{L_{other}}} = c_{i_{L1} j_{L_{other}}} \tag{A.5}$$

$$c^*_{v_{L_{other}} w_{L_{new}}} = c_{i_{L_{other}} j_{L1}} + c_{i_{L_{other}} j_{L2}} \tag{A.6}$$

For the merging of a common label $L1$ with another common label $L2$, the new substitution counts are

$$c^*_{v_{L_{new}} w_{L_{new}}} = c_{i_{L1} j_{L1}} + c_{i_{L2} j_{L2}} + c_{i_{L1} j_{L2}} + c_{i_{L2} j_{L1}} \quad \text{(A.7)}$$

$$c^*_{v_{L_{other}} w_{L_{new}}} = c_{i_{L_{other}} j_{L1}} + c_{i_{L_{other}} j_{L2}} \quad \text{(A.8)}$$

$$c^*_{v_{L_{new}} w_{L_{other}}} = c_{i_{L1} j_{L_{other}}} + c_{i_{L2} j_{L_{other}}} \quad \text{(A.9)}$$

For the merging of a genuine-only label $L1$ with a classified-only label $L2$, the new substitution counts are

$$c^*_{v_{L_{new}} w_{L_{new}}} = c_{i_{L1} j_{L2}} \quad \text{(A.10)}$$

$$c^*_{v_{L_{new}} w_{L_{other}}} = c_{i_{L1} j_{L_{other}}} \quad \text{(A.11)}$$

$$c^*_{v_{L_{other}} w_{L_{new}}} = c_{i_{L_{other}} j_{L2}} \quad \text{(A.12)}$$

A.4 Experiments using Merged Labels

X1(EP): EER			
PGn	**4 CS**	**8 CS**	**16 CS**
Original	13.46%	10.99 %	9.58 %
PG34	15.80%	11.16 %	10.41 %
PG29	16.98%	12.69 %	11.53 %
PG24	17.80%	13.74 %	13.12 %
PG19	17.15%	13.98 %	13.21 %
PG14	18.45%	14.95 %	14.41 %
PG9	25.73%	23.99 %	22.68 %

Table A.1: ***First-Order (X1)*** *EERs of the different merged feature sets. Initialisation with* ***equal probabilities (EP)****.*

X2(EP): EER			
PGn	**4 CS**	**8 CS**	**16 CS**
Original	14.53%	14.95 %	12.18 %
PG34	17.28%	13.58 %	9.95 %
PG29	21.74%	14.54 %	11.90 %
PG24	21.80%	16.01 %	13.11 %
PG19	20.41%	15.77 %	14.32 %
PG14	20.10%	17.13 %	14.50 %
PG9	26.11%	23.51 %	23.25 %

Table A.2: ***Second-Order (X2)*** *Equal Error Rates (EER) of the different merged feature sets. Initialisation with* ***equal probabilities (EP)****.*

A.5 Merged Models as Prior Models

A.5.1 Concept and Approach

Generation of an Equivalent Unmerged Model

Let us define the feature set from which labels in a merged feature set was merged as the **unmerged** feature set. The model computed from the **merged** feature set is a **merged** model, and the model computed from the **unmerged** feature set is an **unmerged** model.

Using a merged model as prior model, we first need to generate an equivalent unmerged model: Each state associated with a merged label is split up into the number of states associated with the labels from which the merged label was merged. A new set of transition probabilities is calculated for these unmerged states. If we divide the transition probabilities evenly among the unmerged states, we can apply the following simple rule: Let state i be a state associated with a label that is not merged, and state j a state associated with a merged label merged from N labels, and a_{ij} the transition probability between state i and j. The new transition probabilities from state i to the N new states, states $j(n)$ $(n = 1..N)$ are

$$a'_{ij(n)} = a_{ij}/N$$

The transition probabilities from state $j(n)$ to state i are

$$a'_{j(n)i} = a_{ji}$$

Figure A.9 is a simple representation of the above process, with $N = 2$ (N being the number of labels that has been merged to a new label). The picture to the right is the equivalent unmerged model of the merged model (left) with label $l_2|l_3$ unmerging to labels l_2 and l_3. Q is the states.

Fusing models

The probabilities of an equivalent model can no be fused with another model trained with the unmerged feature set.

Take a feature set F with $n = 1, 2..N$ phoneme labels. Let $M1$ be an unmerged equivalent model, having an unmerged feature set F, and $M2$ be another model

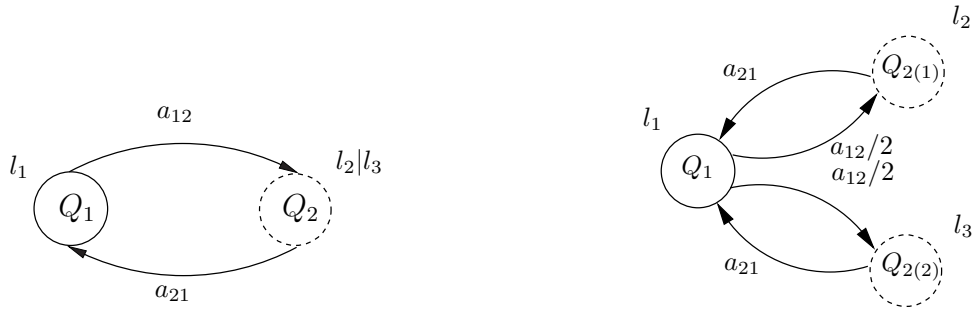


Figure A.9: *The equivalent unmerged model (right) of a merged model (left), with state 2 being a state with merged label $l_2|l_3$, merged from labels l_2 and l_3.*

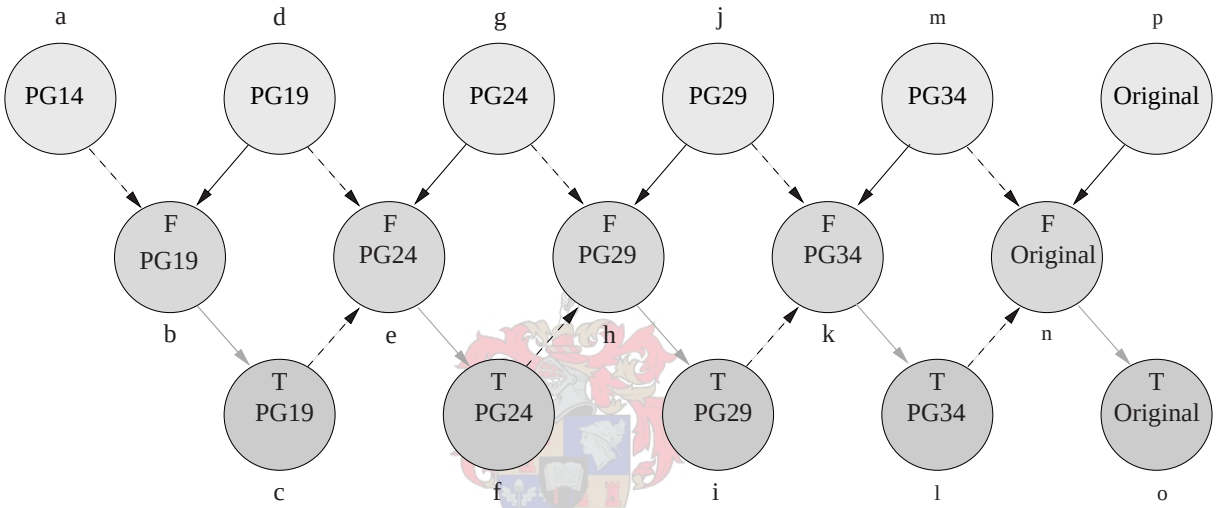


Figure A.10: *A representation of the fusion and training of unmerged equivalent models. The dashed arrows illustrate the computation of equivalent unmerged models.*

generated from F. Let $p1(n)$ be the n-th transition probability of a specific state of $M1$ and $p2(n)$ the n-th transition probability of the same state of $M2$. (The index n is linked to state n associated with the $n-th$ phoneme). A new fused model with transition probabilities $p3$ can be generated by adding the probabilities of $M1$ and $M2$ as follows:

$$x3(n) = \beta * p1(n) + (1-\beta) * p2(n) \qquad n = 1, 2, \ldots N \tag{A.13}$$

where N is the number of phoneme labels in the feature set, β is a fusion probability, and $x3$ is transition values that are normalised to get $p3$:

$$p3(n) = x3(n) / \textstyle\sum_n x3(n) \qquad n = 1, 2, \ldots N \tag{A.14}$$

It is now possible to train yet another model by smoothing it with the fused model $M3$.

There are a number of fusion possibilities of the various models as illustrated in Figure A.10. The circles at the first (highest) level each represent a model computed

from different merged feature sets. For instance, the circle with $PG14$, represents a model computed from the merged feature set with 14 phoneme labels in total. The circles at the second level, all marked with the symbol F, represent models that are generated by fusing two other model probabilities, by using the method described by Equation A.14. The circles at the third (lowest) level represent models that are further trained by using the fused models in the second level as prior models. The dashed arrows represent the generation of equivalent unmerged models, while the grey arrows represent the training by means of smoothing fused models.

To illustrate: the equivalent unmerged model of model $PG14$ in Figure A.10 is fused with model $PG19$ to generate a fused model $FPG19$ with a transition probability matrix A_{fuse}. We further train the transition probabilities of $FPG19$, smoothing each state with its corresponding probabilities of A_{fuse}, finally generating model $TPG19$ at the third level.

To simplify explanation of these different paths, let us replace the representation of Figure A.10 with the one shown in Figure A.11, where the symbols match those of Figure A.10. For our experiments in this chapter, the unfused merged models at the top level of Figure A.10 (represented by the symbols at the top level of Figure A.11) are initialised by using normalised bigram counts (BG).

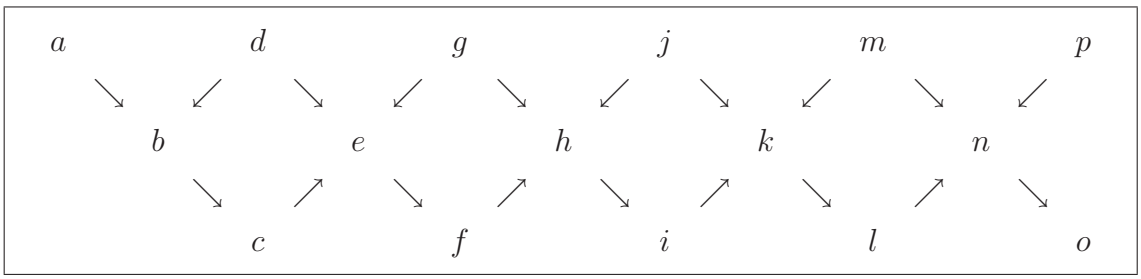


Figure A.11: *A compact simplification, replacing the fusion and training structure of that of Figure A.10.*

By making use of the structure of Figure A.11, we experiment choosing two main routes of this structure.

Route A

The first route is shown in Figure A.12. The following rule is applied to experiments that use this route: After a fused model (second level symbols) is trained and smoothed to eventually generate a trained model (third level symbol), the trained model is never again used for fusion. The trained models at the third level are then used to generate speaker verification results.

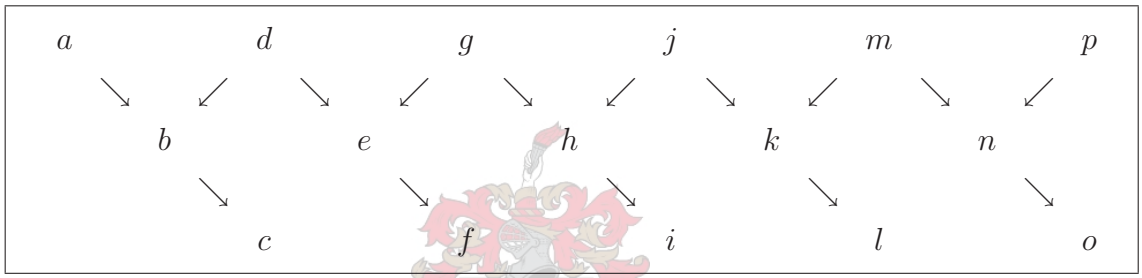


Figure A.12: *The structure of experiments taking Route A described in Section A.5.1*

We experiment using different fusion probabilties (β values) in Equation A.14. Two types of verifiers are used taking Route A: One is where the impostor models are trained, using exactly the same structure as in Figure A.12 up to the third level. The imposter models are fused and trained using the same route and the same fusion probability, β, as the target models. The other verifier type is where the impostor models are kept unfused as in the top level of Figure A.12.

Route B

The second route that is used for our experiments allows us to fuse models at the third level of Figure A.10. First we generate a fused model $FPG19$ from the equivalent unmerged model of model $PG14$ and model $PG19$, and smooth and train $FPG19$ to finally generate model $TPG19$, as shown in Figure A.13.

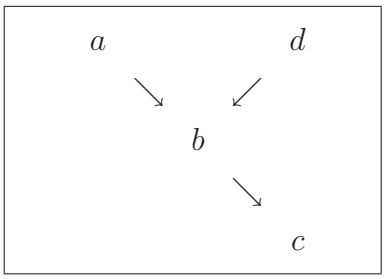


Figure A.13: *An illustration of the first step of the structure of Route B described in Section A.5.1. This involves the fusion of $PG19$ and the unmerged equivalent of model $PG14$ and the smoothing and training of the fused model $FPG19$ of Figure A.10.*

The next fused model $FPG24$ we generate, is fused from the unmerged equivalent of model $TPG19$ and model $PG24$, and then smoothed and trained to form a structure such as the one in Figure A.14. This is repeated until the total path that we followed for fusion, smoothing and training of Route B looks like the one in Figure A.15.

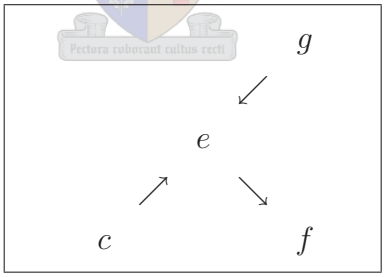


Figure A.14: *An illustration of the second step of the structure of Route B described in Section A.5.1. This involves the fusion of the unmerged equivalent of the trained model $TPG19$ at the third level of Figure A.10 and of model $PG24$. The resultant fused model $FPG24$ is smoothed and trained to generate model $TPG24$ in Figure A.10.*

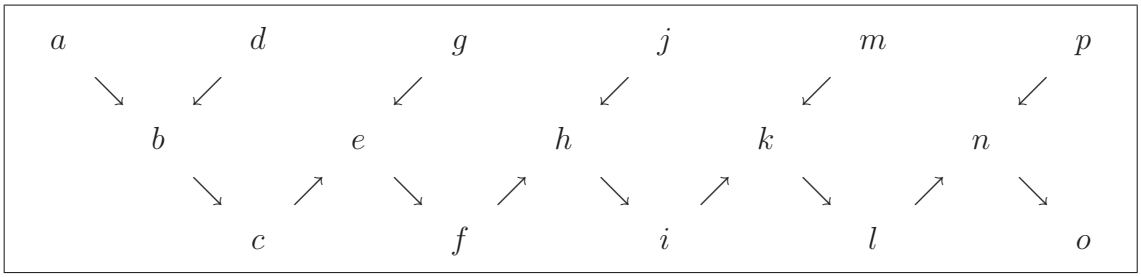


Figure A.15: *The total structure of Route B, created after following all the options with the different feature sets. Comparing this with Figure A.10, this figure shows that this structure allows fusion of trained models at the third level.*

Experiments taking Route B are done by using unfused impostor models at the top level of Figure A.10. In other words, we do not train the impostor models in the same way as we do the target models. This is done to save time and to simplify things.

A.5.2 Experiments

In Section 4.2.1 the generation of merged models containing fewer parameters, was discussed. This was attained by merging labels that cause the most confusion. In Section A.5 the use of merged models as prior models was discussed. We described how an unmerged equivalent model can be generated and how these models are fused and trained. Two basic structures defined as Route A and Route B, which are used for fusion and training were discussed. The same set of 40 impostor speakers are chosen as in Chapters 3 and 4.

Tables A.3 to A.6 show the EER results using Route A of Section A.5.1. A prior factor of $\alpha = 20$ is used. A fusion factor of $\beta = 0.1$ (refer to Equation A.14) is used in the results of Tables A.3 and A.4, while a fusion factor of $\beta = 0.3$ is used in the results of Tables A.5 and A.6. The fusion factors are chosen relatively small, in order to keep the influence of the prior model relatively low. Initialisation is done with BG.

We used two types of verifiers:

a One that uses unfused impostor models. This is used in the experiments of Tables A.3 and A.5.

b One that fuses and smooths the impostor models in exactly the same way as the speaker models. This is used in the experiments of Tables A.4 and A.6.

X1_Smooth_Via_Merged_Symbols: EER $\alpha = 20$, $\beta = 0.1$			
PG*n*	**4 CS**	**8 CS**	**16 CS**
Original_a	13.88%	10.75 %	9.68 %
PG34_a	15.95%	11.97 %	11.17 %
PG29_a	16.31%	12.20 %	12.36 %

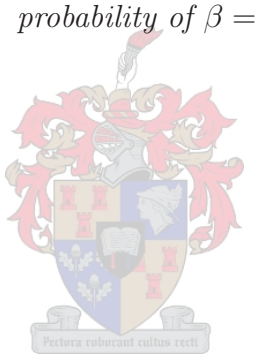
Table A.3: *The EER results using the transition probabilities of* ***equivalent unmerged models*** *as priors, with a Dirichlet prior factor* $\alpha = 20$ *and a fusion probability of* $\beta = 0.1$.

X1_Smooth_Via_Merged_Symbols: EER $\alpha = 20$, $\beta = 0.1$			
PG*n*	**4 CS**	**8 CS**	**16 CS**
Original_b	14.31%	10.91 %	9.77 %
PG34_b	15.66%	11.31 %	10.87 %
PG29_b	15.98%	13.49 %	13.76 %

Table A.4: *The EER results using the transition probabilities of* ***equivalent unmerged models*** *as priors, with a Dirichlet prior factor* $\alpha = 20$ *and a fusion probability of* $\beta = 0.1$.

X1_Smooth_Via_Merged_Symbols: EER $\alpha = 20$, $\beta = 0.3$			
PG*n*	**4 CS**	**8 CS**	**16 CS**
Original_a	14.11%	10.83 %	9.40 %
PG34_a	16.25%	11.88 %	10.60 %
PG29_a	16.11%	12.45 %	11.81 %

Table A.5: *The EER results using the transition probabilities of* ***equivalent unmerged models*** *as priors, with a Dirichlet prior factor* $\alpha = 20$ *and a fusion probability of* $\beta = 0.3$.



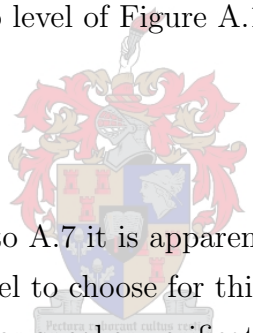
X1_Smooth_Via_Merged_Symbols: EER $\alpha = 20$, $\beta = 0.3$			
PG*n*	**4 CS**	**8 CS**	**16 CS**
Original_b	14.25%	10.76 %	9.66 %
PG34_b	16.11%	11.40 %	10.69 %
PG29_b	16.83%	12.61 %	11.62 %

Table A.6: *The EER results using the transition probabilities of* ***equivalent unmerged models*** *as priors, with a Dirichlet prior factor* $\alpha = 20$ *and a fusion probability of* $\beta = 0.3$.

X1_Smooth_Via_Merged_Symbols (Chain): EER $\alpha = 20$, $\beta = 0.2$			
PGn	**4 CS**	**8 CS**	**16 CS**
Original	13.72%	10.75 %	9.11 %
PG34	16.18%	11.31 %	11.06 %
PG29	16.16%	12.69 %	11.99 %

Table A.7: *The EER results using the transition probabilities of* ***equivalent unmerged models*** *as priors, with a Dirichlet prior factor* $\alpha = 20$ *and a fusion probability of* $\beta = 0.2$.

Table A.7 shows the EER results using Route B of Section A.5.1. A fusion probability of $\beta = 0.2$, and a prior factor of $\alpha = 20$ are used. The impostor models are not trained using the same structure as that of the target models. We keep the impostor models unfused as illustrated in the top level of Figure A.10.



Conclusion

From the results of Tables A.3 to A.7 it is apparent that choosing a merged model as prior model is not a fitting model to choose for this goal. This procedure is complex and expensive and is of little value for speaker verification. The idea of merging phoneme labels that cause the most confusion seemed like a valid experiment at the time as the aim was to explore the effects of using fewer parameters for speaker verification. It would appear that, by merging the labels 5 at a time, the categories too soon become broadly defined to be of any value for speaker verification. The merged models are already of such insignificant value for speaker verification, that using them as prior models to smooth other models would not help for speaker verification.

A.6 Experiments Using Merged Models in Combination with Other Smoothing Techniques

A.6.1 Combined with Uniform Smoothing

X1_UP(BG): EER			
PG*n*	**4 CS**	**8 CS**	**16 CS**
PG34	15.49%	11.48 %	11.06 %
PG29	16.85%	12.12 %	12.18 %
PG24	17.96%	14.21 %	12.83 %
PG19	17.43%	13.50 %	13.85 %
PG14	19.89%	16.33 %	15.72 %

Table A.8: *The EERs of experiments using the merging of labels in combination with using* ***Uniform Priors, smoothing first-order*** *HMM speaker models (X1_UP). A prior factor of 5 is used and initialisation is done with* ***normalised bigram counts (BG)***

X2_UP(EP): EER			
PG*n*	**4 CS**	**8 CS**	**16 CS**
PG34	18.70%	13.01 %	9.95 %
PG29	22.70%	15.75 %	13.67 %
PG24	21.22%	16.73 %	14.68 %
PG19	19.91%	17.71 %	17.94 %
PG14	26.78%	28.36 %	28.25 %

Table A.9: *The EERs of experiments using* ***Uniform Priors, smoothing second-order*** *HMM speaker models (X2_UP). A prior factor of 5 is used and initialisation with* ***equal probabilities (EP)***

A.6.2 Combined with Dirichlet Smoothing

X2_DP(EP): EER $\alpha = 40$			
PGn	**4 CS**	**8 CS**	**16 CS**
PG34	15.49%	11.65 %	9.57 %
PG29	17.52%	12.41 %	11.05 %
PG24	19.04%	15.09 %	13.74 %
PG19	16.96%	14.16 %	14.13 %
PG14	18.85%	16.72 %	15.62 %

Table A.10: *The EERs of experiments using the merging of labels in combination with using first-order TSMs as prior models for smoothing second-order TSMs (X2_DP). Initialisation with* ***equal probabilities (EP)***

A.7 Observation Counts

Total number of observation counts, $\sum_{\mathbf{i}} \mathbf{c_i}$			
HMM Order	**4 CS**	**8 CS**	**16 CS**
1st	60 – 300	100 – 600	200 – 1200
2nd	25 – 60	30 – 75	40 – 110

Table A.11: *Typical ranges of the total number of observation counts in a state of a speaker HMM, using NIST Switchboard II.*

A.8 Significant Probability Levels and DET curves

A.8.1 First-Order Experiments with and without Smoothing

$X1_A$**:** This is a first-order result where no smoothing is used.

$X1_B$**:** This is a result where a UBM is used as prior model to smooth first-order speaker HMMs. A Dirichlet estimator is used with $k_\alpha = 1$ (Equation 5.2), and $k_\lambda = 5$ (Equation 5.1).

The same model structure, initialisation and verification are used as in the first-order results of Chapters 3 and 4. (Both $X1_A$ and $X1_B$ use the same first-order impostor set for verification.) The results of Figures A.16 to A.18 show that over the widest range of the DET curve, $X1_B$ does not perform significantly better than $X1_A$. This suggest that there is sufficient data for first-order speaker modelling. These results were generated by using the McNemar significance test and implementing it as described in Section 5.4

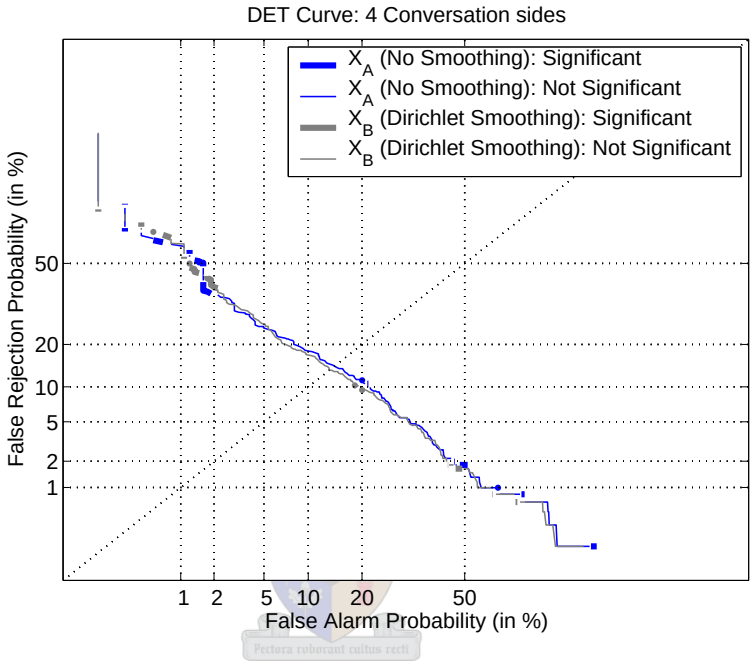


Figure A.16: *Significant probability levels plotted against $r = FRR/FAR$ where X_A performs better than X_B, using 4 conversation sides for training.*

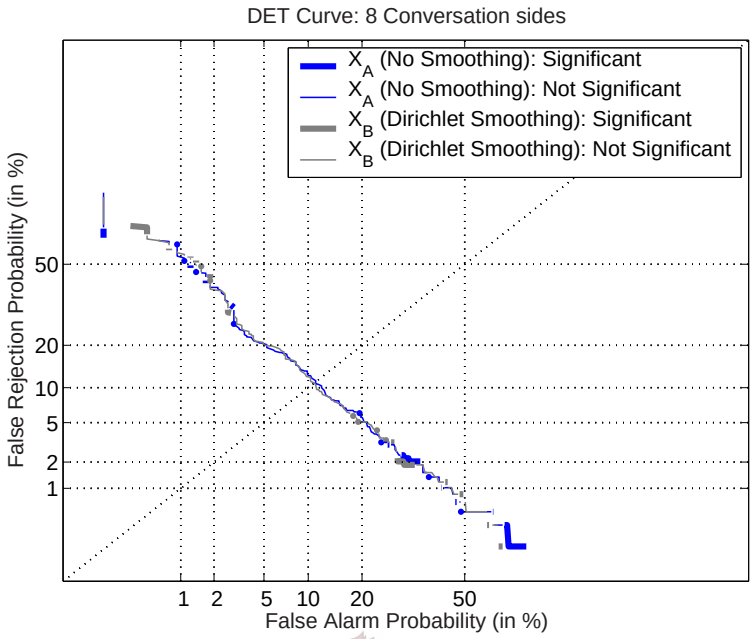


Figure A.17: *Significant probability levels plotted against $r = FRR/FAR$ where X_A performs better than X_B, using 8 conversation sides for training.*

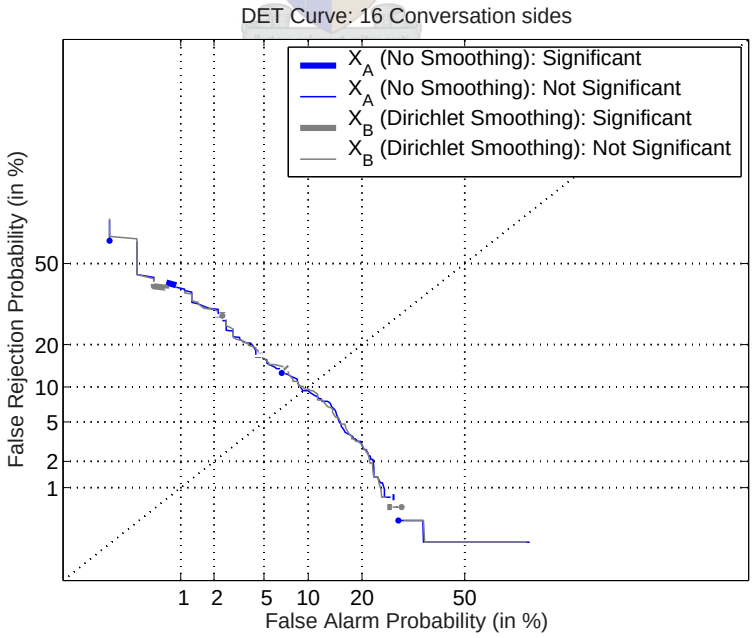


Figure A.18: *Significant probability levels plotted against $r = FRR/FAR$ where X_A performs better than X_B, using 16 conversation sides for training.*

A.8.2 First-Order and Second-Order Experiments

X1 is a first-order result without using any smoothing. X2 is a second-order result, using first-order target models as prior models for the second-order target models.
Figures A.21 to A.23 are plots of the significance probability vs $r = FRR/FAR$ at areas where X2 has performed better than X1. The dashed line indicates a significance level of $\epsilon = 0.1$, where if $P < \epsilon$, X2 has performed significantly better than X1. Above this level the difference is not significant.

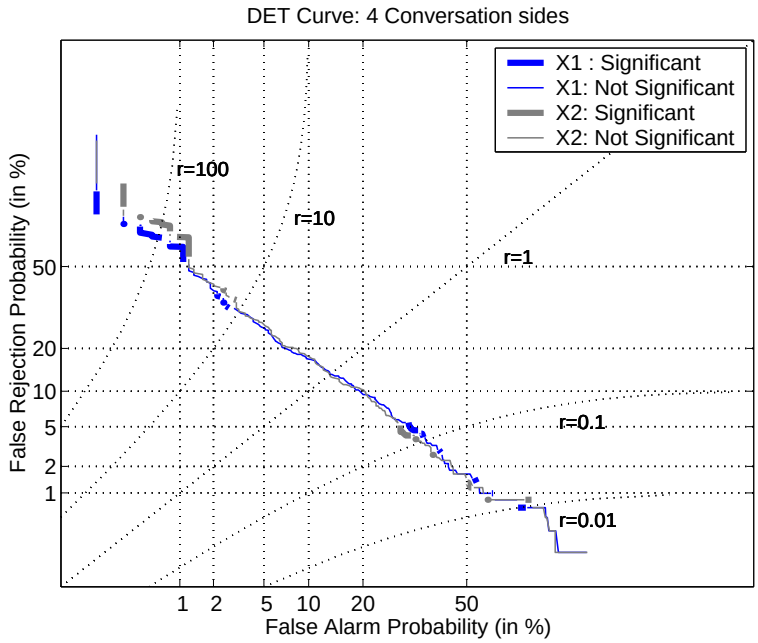


Figure A.19: *DET curve of X1 and X2 using 4 conversation sides for training, indicating the significant difference with thicker curves*

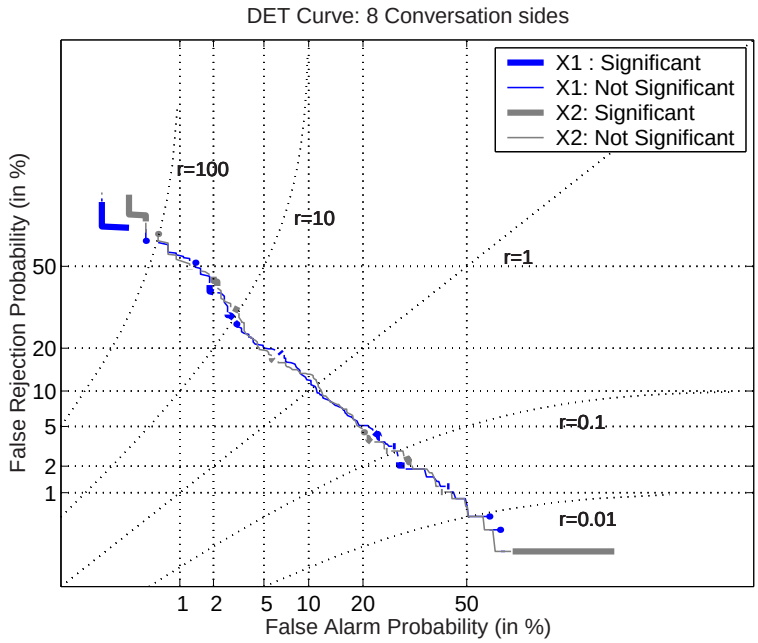


Figure A.20: *DET curve of X1 and X2 using 8 conversation sides for training, indicating the significant difference with thicker curves*

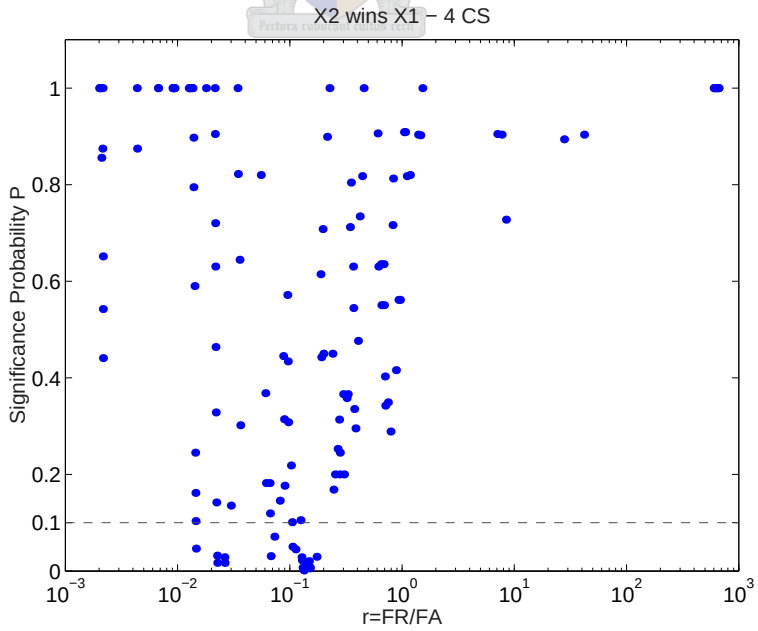


Figure A.21: *Significance Probability P vs $r = FRR/FAR$, where X2 performs better than X1, using 4 conversation sides for training.*

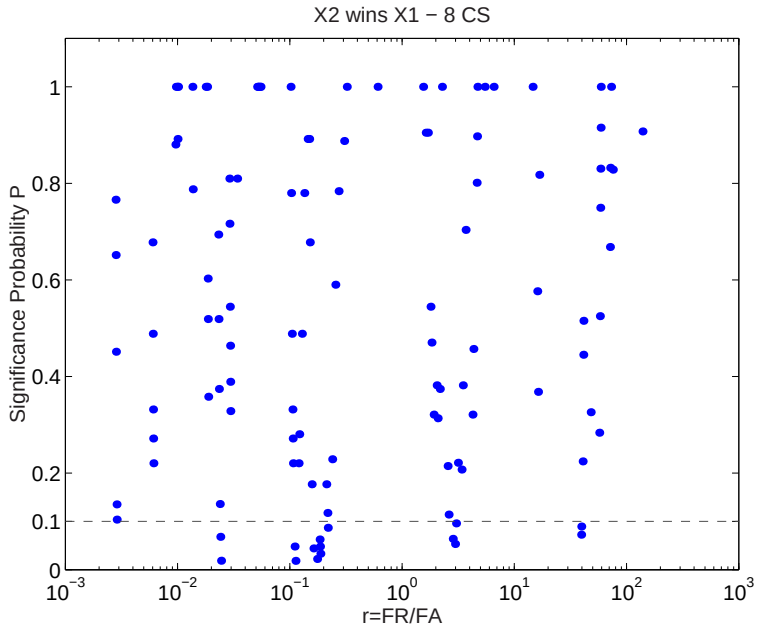


Figure A.22: *Significance Probability P vs $r = FRR/FAR$, where X2 performs better than X1, using 8 conversation sides for training.*

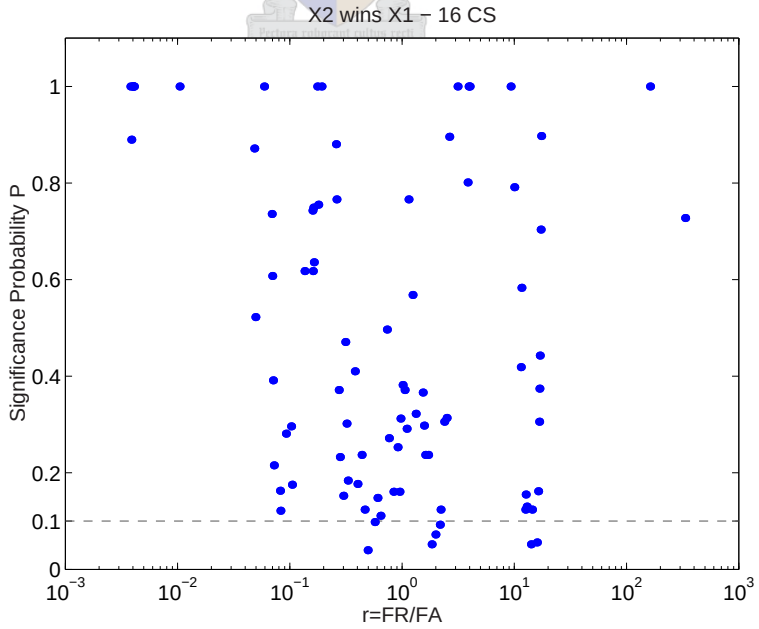


Figure A.23: *Significance Probability P vs $r = FRR/FAR$, where X2 performs better than X1, using 16 conversation sides for training.*

Appendix B

Lexical Speaker Recognition

B.1 Influence of the Minimum word count

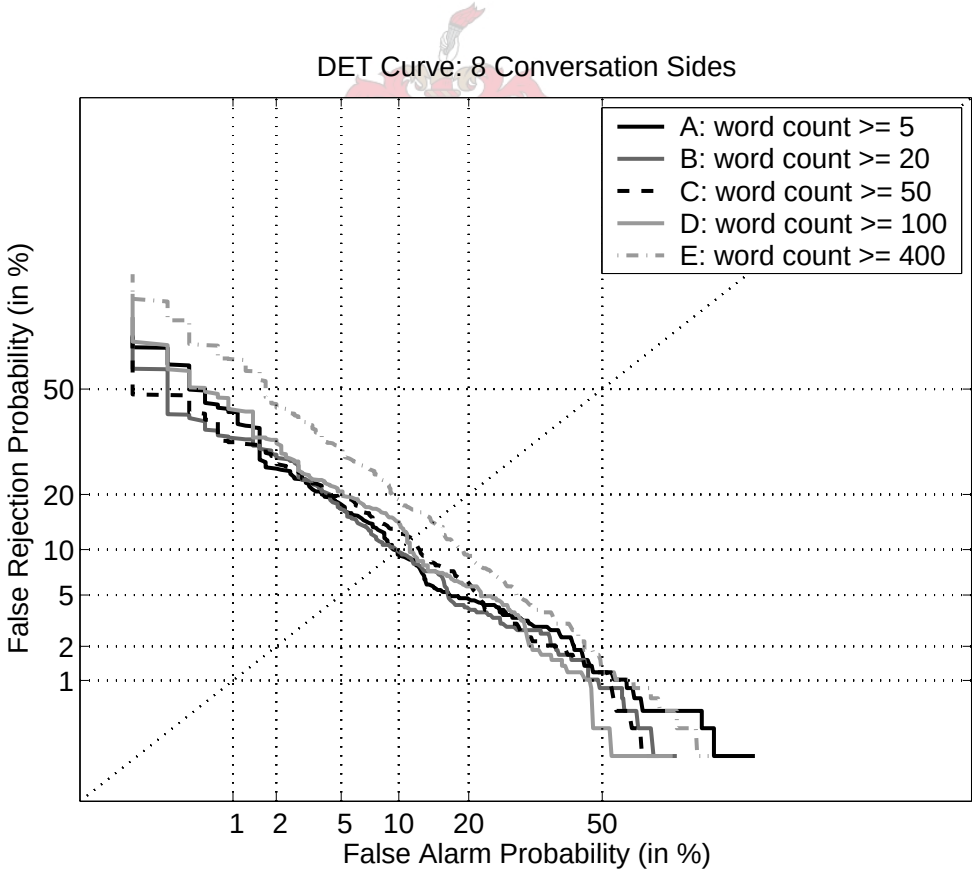


Figure B.1: *DET curves of feature sets containing words that are selected based on minimum word counts, using 8 conversation sides for training.*

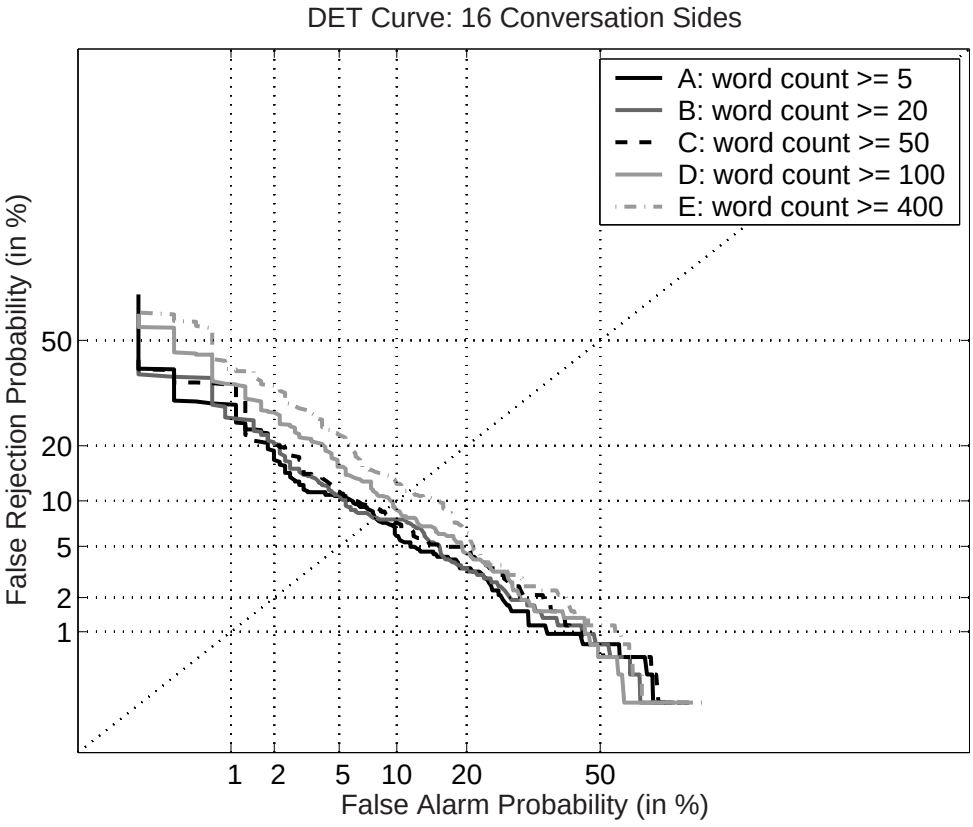


Figure B.2: *DET curves of feature sets containing words that are selected based on minimum word counts, using 16 conversation sides for training.*

B.2 Influence of the Maximum word count

The effect of excluding words with the highest word count also needs investigating. To do this, we set up 4 feature sets, each having a minimum word count of 100, where each excludes words with the highest word count:

***F*:** excludes words with the top 10 highest word counts,

***G*:** excludes words with the top 50 highest word counts,

***H*:** excludes words with the top 150 highest word counts,

***I*:** excludes words with the top 491 highest word counts, giving a total of 1000 words in the feature set.

These four feature sets are used for speaker recognition, together with feature set *D* which has a minimum word count of 100 with none of the top word counts being excluded. Figure B.3 to B.5 shows the DET curves using these features sets. The EERs are shown in Table B.1.

Feature Set	Equal Error Rate (EER)			Overall Accuracy
	4 CS	8 CS	16 CS	
D: word count >= 100	14.19 %	10.99 %	9.66 %	80.23 %
F: excluding top 10 word counts	14.46 %	12.12 %	10.42 %	82.45 %
G: excluding top 50 word counts	18.61 %	14.79 %	12.65 %	82.12 %
H: excluding top 150 word counts	25.83 %	20.29 %	15.90 %	78.35 %
I: excluding top 491 word counts	31.38 %	27.82 %	22.49 %	70.25 %

Table B.1: *EER results of feature sets excluding words with the highest word counts, using 4, 8 and 16 conversation sides (CS) for training.*

Conclusion

The results of feature sets *D* and *F* are comparable, but as we discard more and more of the words with a high word count, the speaker recognition performance drops drastically. Feature set *I*, which discarded words with the top 491 highest word counts, has an overall accuracy as low as 70.25%. The results show that it is better to exclude words with low counts, than to exclude words with high counts, as done in Section 6.6.1.

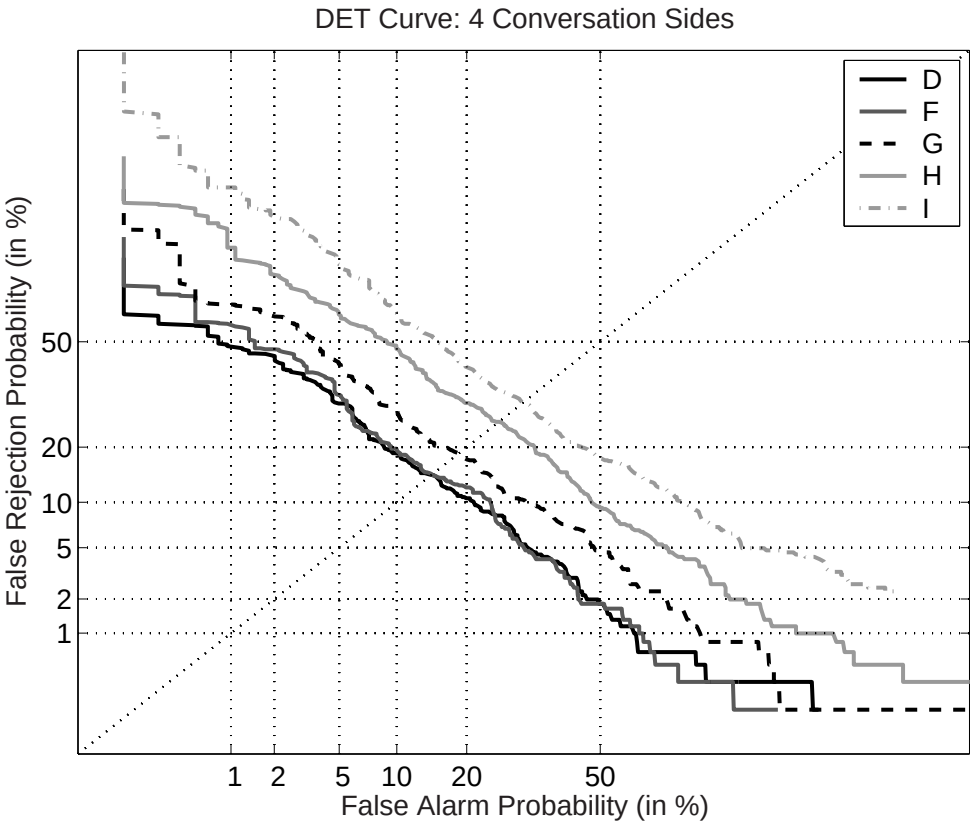


Figure B.3: *DET curves of feature sets excluding words with the highest word counts, using 4 conversation sides for training.*

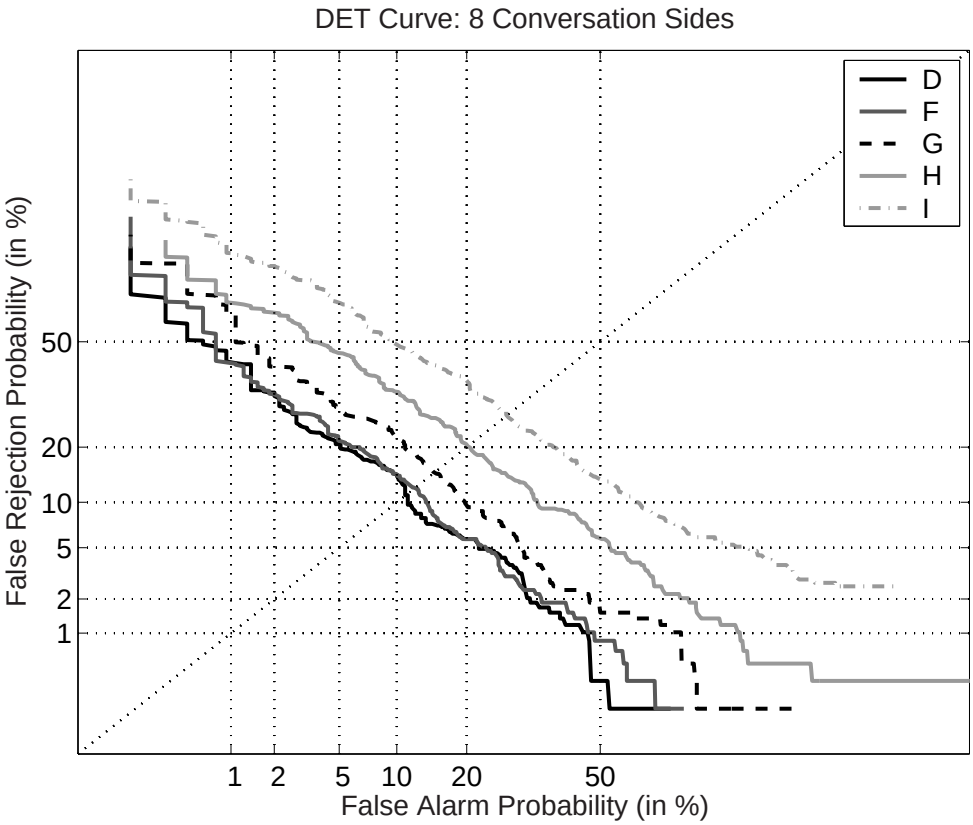


Figure B.4: *DET curves of feature sets excluding words with the highest word counts, using 8 conversation sides for training.*

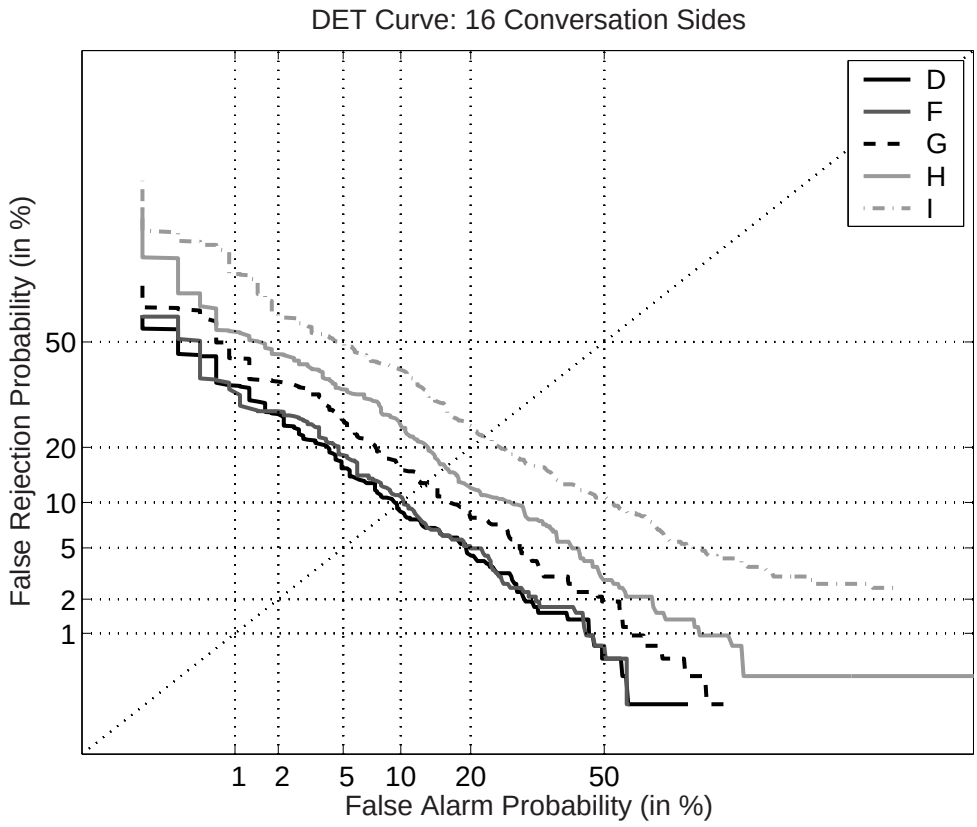


Figure B.5: *DET curves of feature sets excluding words with the highest word counts, using 16 conversation sides for training.*

B.3 Selection of words based upon Entropy

Using the definition of speaker entropy of Equation 6.1, we make a selection of words of which the speaker entropy is the lowest. Section 6.4.2 states that a low speaker entropy indicates words that are valuable in distinguishing amongst speakers.

The following experiments are done, comparing two types of feature sets: The first type selects the N words with the highest word counts and the second type selects the N words with the lowest speaker entropy (see Section 6.4.2), ignoring words with a word count below a chosen threshold. Words below a certain threshold are ignored, because it is highly unlikely that these words appear in more than one of the impostor, training and test set. The number of word labels in a feature set is varied, as well as the minimum word count. Two sets of speakers are used:

- One speaker set contains 602 speakers from the UBM set out of jack_1 (jackknife set 1) to jack_9 (jackknife set 9) in Switchboard II. All the conversation sides from the same speaker are pooled together. The set of 602 speakers has approximately 22 000 unique classified word labels and a total word count of approximately 28 million.
- The other speaker set contains all 31 target speakers from jack_0 in Switchboard II. The conversation sides of the same speaker are pooled together as well. This speaker set has approximately 7000 unique classified word labels and a total word count of approximately 1.6 million. We therefore tend to have a lower minimum word count threshold for this set of speakers than for the set of 602 UBM speakers.

We count the classified word labels in the conversation pool of all the speakers, and count the total number of classified word labels of all the speakers to compute the word probabilities in Equation 6.1. Note that the word selections are made from Switchboard II and not Switchboard I (as is done with the selection of words based upon minimum word count in Section 6.6.1)[1]. We experiment with the following feature sets:

[1]The words are selected from Switchboard II to make it possible to compare results to those obtained by selecting words based on log-probability in Section B.4.

B.3.1 Feature sets

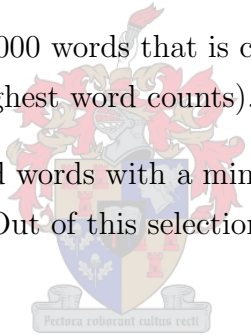
The following feature sets are selected from the word labels in the set of 602 UBM speakers. We choose the set of UBM speakers to make a selection of words that would generally be good words to distinguish amongst speakers and which are not dependent on the target speaker set.

C1: This is a feature set that contains 4000 classified word labels with the highest word count.

E1: The number of different speakers for which a specific word is classified is counted. Let us refer to this count as the speaker count. We first select the classified words with a minimum word count of 30 and a speaker count of over 10. (A selection of 5 631 words is made.) Out of this selection of words we choose the 4000 words with the lowest speaker entropy.

C2: This feature set contains 6000 words that is classified the most. (The classified words with 6000 of the highest word counts).

E2: We first select the classified words with a minimum word count of 30. (A selection of 8 153 words is made.) Out of this selection we choose the 6000 words with the lowest speaker entropy.



The following feature sets are selected from the word labels in the set of 31 target speakers and are therefore dependent on the set of target speakers.

C3: This feature set contains 2000 classified word labels with the highest word count.

E3: There are 4 029 classified words with a minimum word count of 8. From these words, 2000 words with the lowest speaker entropy are selected.

C4: This feature set contains 3000 classified word labels with the highest word count.

E4: There are 2 723 classified words with a minimum word count of 15. From this selection we choose the 3000 words with the lowest speaker entropy.

In the selection of Feature set *E2*, the word "ergonomics" appeared 96 times, but was used by only one speaker. This word did not appear in the set of target speakers. Therefore, in a selection of words that should be independent of the target set, it makes

more sense to count the number of times a word is used by different speakers and to select words which are used by more than a chosen number of speakers, as is the case with Feature set *E1*. This would make the probability of a word appearing in the target set slightly higher.

The selection of words based on speaker entropy is a complex problem and needs a great deal of data. This selection fares much worse than the selection based on word counts, as we have too little data available in determining which selection of words is useful to distinguish amongst a group of speakers. The results are shown in Figures B.6 to B.9. Table B.2 is a summary of the EERs of the DET curves in Figures B.6 to B.9 (Using feature sets C1 to C4 and E1 to E4).

Feature Set	Equal Error Rate (EER)		
	4 CS	8 CS	16 CS
C1:	12.55 %	9.86 %	6.88 %
E1:	16.33 %	11.24 %	9.30 %
C2:	12.83 %	8.97 %	7.53 %
E2:	16.64 %	11.17 %	8.92 %
C3:	11.51 %	10.50 %	8.74 %
E3:	15.13 %	12.28 %	9.85 %
C4:	11.80 %	9.54 %	7.44 %
E4:	14.93 %	11.96 %	9.58 %

Table B.2: *EERs using the feature sets described in section B.3 of which the DET curves are shown in Figures B.6 to B.9.*

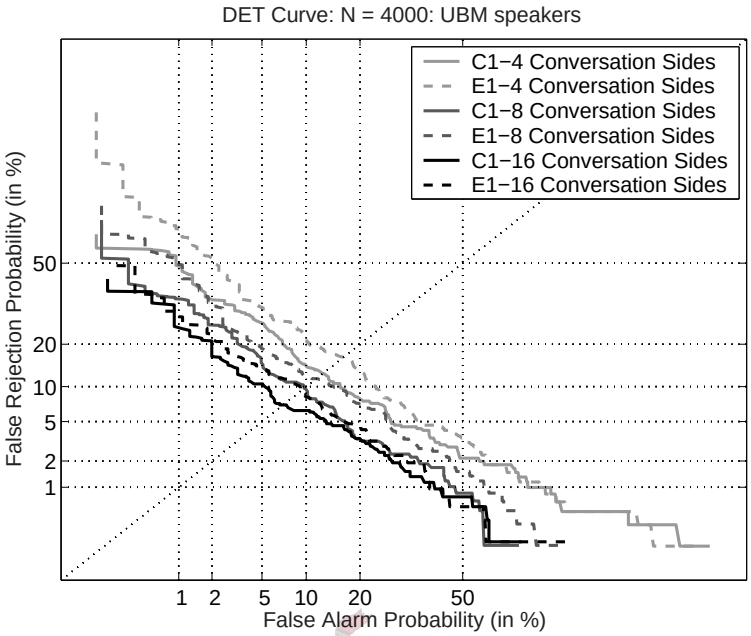


Figure B.6: *DET curves of experiments using a feature set containing 4000 unique word labels: One generated using the speaker entropy of 641 UBM speakers and the other using the 4000 words with the highest word count out of the same set of speakers.*

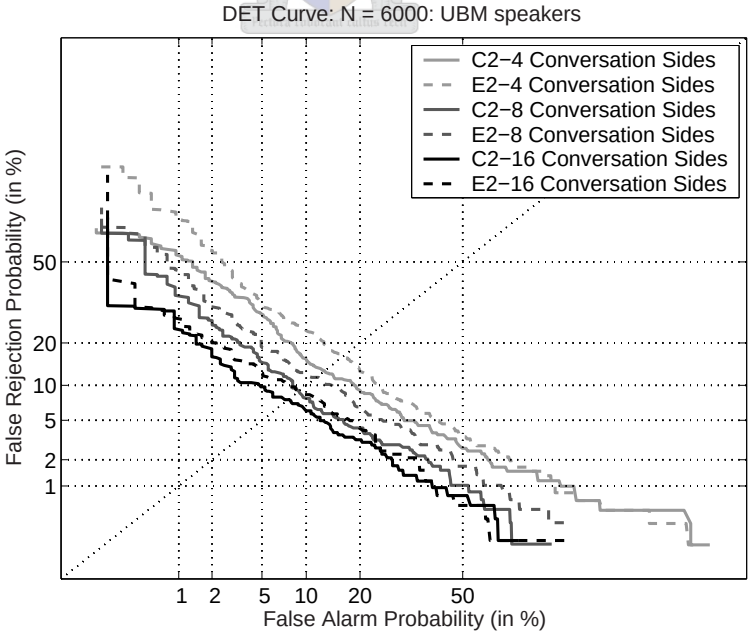


Figure B.7: *DET curves of experiments using a feature set containing 4000 unique word labels: One generated using the speaker entropy of 641 UBM speakers and the other using the 6000 words with the highest word count out of the same set of speakers.*

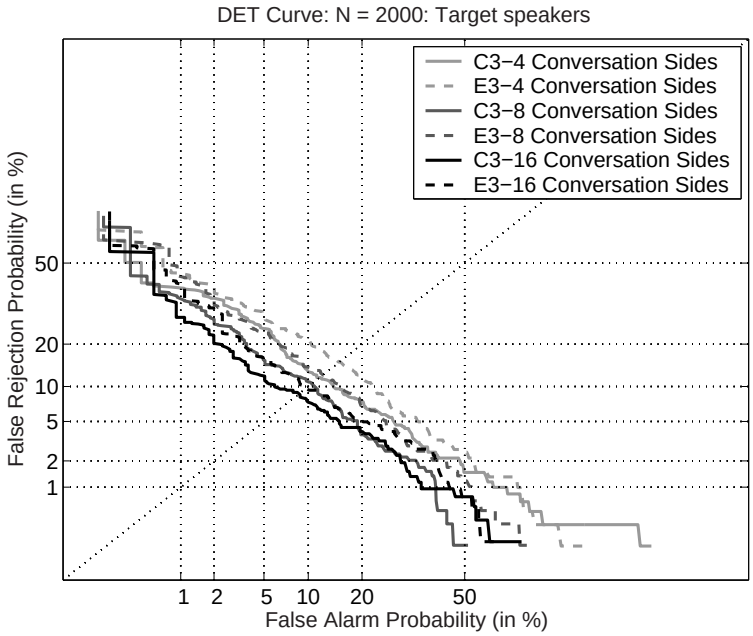


Figure B.8: *DET curves of experiments using a feature set containing 2000 unique word labels: One generated using the speaker entropy of 31 target speakers and the other using the 2000 words with the highest word count out of the same set of speakers.*

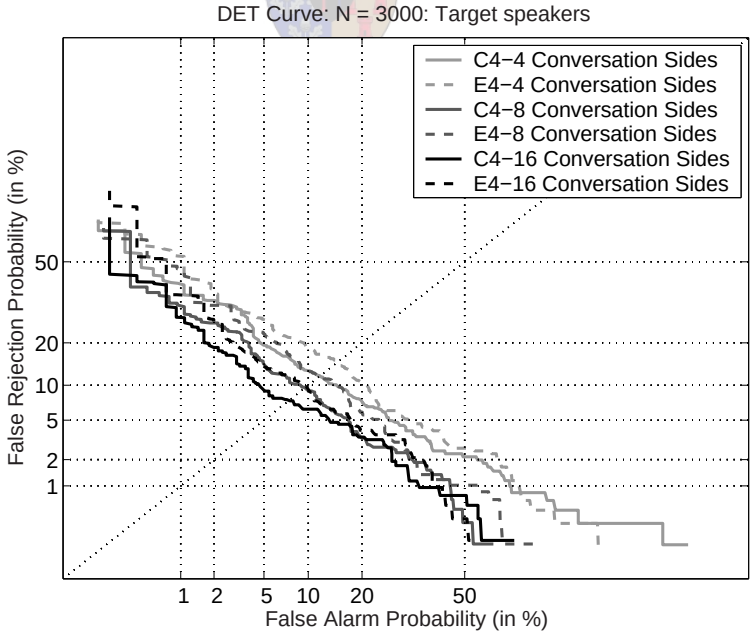


Figure B.9: *DET curves of experiments using a feature set containing 3000 unique word labels: One generated using the speaker entropy of 31 target speakers and the other using the 3000 words with the highest word count out of the same set of speakers.*

B.4 Selection of words based upon Log-probabilities

In Section 6.4.3 we explain how we can determine which words a target speaker uses that are used very differently from the average speaker. This could mean that the target speaker uses the word significantly more or significantly less than the average speaker. We determine this by evaluating the difference of the log-probabilities of words from Switchboard II of the UBM model and those from target models in jack_0 (jackknife set 0). The words are selected from Switchboard II, since we would like to compare words used by the average speaker (represented by the set of UBM speakers). The set-up of the experiments in this section can be divided into 4 steps:

Step 1 The target models are trained with a feature set containing all the words in Switchboard II, not using a UBM prior model to smooth the target models. The target models are smoothed by adding a constant value (1 divided by the number of word labels in Switchboard II) to all the word probabilities, and then normalising the probalities. (This ensures no numerical error in the calculation of the log-probability of a word that does not appear in the target set).

Step 2 A UBM model is trained using all the words in the Switchboard II corpus. The probabilities of the words are smoothed by adding a constant factor of 1 divided by the number of word classes in Switchboard II to them.

Step 3 The feature sets for each target model are generated containing target-specific words. The deviations in Equation 6.3 are evaluated.

Words that are used more often or equally often by the average speaker (represented by the UBM) than the target speaker are selected as follows:

- These are words with a deviation $d(n, i) >= 0$. (The deviation $d(n, i)$ is a function of the n-th word and i-th target model.)
- Words in this category are excluded from the feature subset if the word count < 50 (in the entire Switchboard II set).

Words that are used more often by the target speaker than the average speaker are selected as follows:

- These are words with a deviation less than zero
- For a word with a word count smaller than 10 in the entire Switchboard II set:
  - If $-6 < d(n,i) < 0$, the word is not selected.
  - All words with a deviation below -6 are selected.

The 2 774 words that are selected as mentioned above included in the feature set. This is done by selecting the 2 774 words where the absolute value of the deviation is the highest. (These are words that are used more often or less often by the target speaker than the average speaker.)

Step 4 Each target model has its own unique feature set containing 2 774 words associated with it. Each of the target models is then retrained using its associated feature set.

Two sets of experiments are done using the target-specific feature sets. One is done where the target model is smoothed using a UBM prior model trained with the same feature set as with the target model, using a prior factor of 0.1. The other experiment does not use a UBM prior model to smooth the target model. In the latter case we smooth the target models by adding $\frac{1}{2774}$ to their word probabilities. The test sequences are verified using a feature set associated with a specific target model and with impostor models trained with the same associated feature set as the target model in question. Figure B.10 shows the DET curves of the cases with and without the use of a UBM prior model. *LP_I* is the experiment not using a UBM prior model and *LP_II* the experiment using a UBM prior model. [2]

The EERs are given in Table B.3. The results are compared to results using feature set C1. Feature set C1 is selected from Switchboard II and contains the 4000 classified word labels with the highest word count. From the results in Table B.3 it is clear that, with the available data this problem is too complex to give adequate results, and we end up

[2]With a selection of words based on log-probabilities of the UBM and the target speaker, it is found that using a UBM prior model is less advantagous than not using it. Most of the words of this selection are not used only by the target speaker and most of these words' probabilities are higher in the UBM than in the target model. Because of the smaller amount of data available when choosing a smaller feature set, the UBM carries more weight in the Dirichlet MAP adaptation of the word probablities than the target model, than it would with a bigger feature set and subsequently more available data.

with the same problem experienced in the case of speaker entropy in Section B.3 (we lack sufficient data).

Other possibilities have been explored, such as the exclusion of words in the target set below a minimum word count of 5. All words with an absolute deviation above a threshold of, say 0.5, are chosen. This generates feature sets with sizes ranging typically from a 100 to 400 for the different conversation sides. These are too small to cover enough words for training of target models, and result in poor accuracies. These feature sets result in a large group of words used mainly by the target speaker and are most likely not to appear in the set of words used by an impostor speaker.

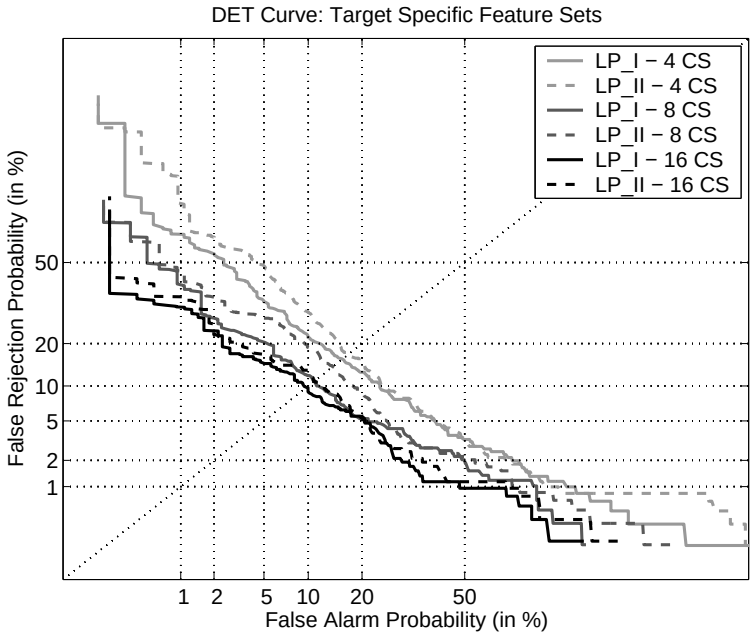


Figure B.10: *DET curves of experiments using feature sets (with 2 774 words) computed from log-probabilities using the different conversation sides (CS). LP_I uses no prioring and LP_II uses UBM prioring.*

Feature Set	Equal Error Rate (EER)		
	4 CS	8 CS	16 CS
LP_I	15.80 %	10.99 %	9.67 %
LP_II	17.23 %	13.98 %	10.88 %
C1:	12.55 %	9.86 %	6.88 %

Table B.3: *The results when using target-specific feature sets ascociated with each target model, generated by evaluating log-probabilities of the UBM and target models, using no prioring (LP_I) and UBM prioring (LP_II). This is compared with the best result of experiments based on word counts of section 6.6.1, using feature set B containing words with a minimum word count of 20.*

B.5 Training Genuine Word Labels

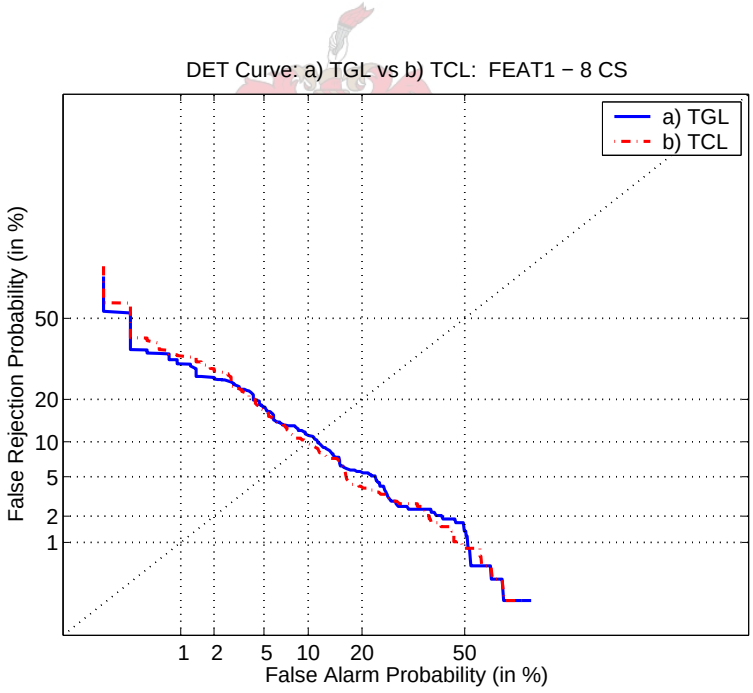


Figure B.11: *Comparison of DET curves of experiments that trained genuine labels (TGL) and experiments that trained with classified labels (TCL) using 8 conversation sides and feature set FEAT1.*

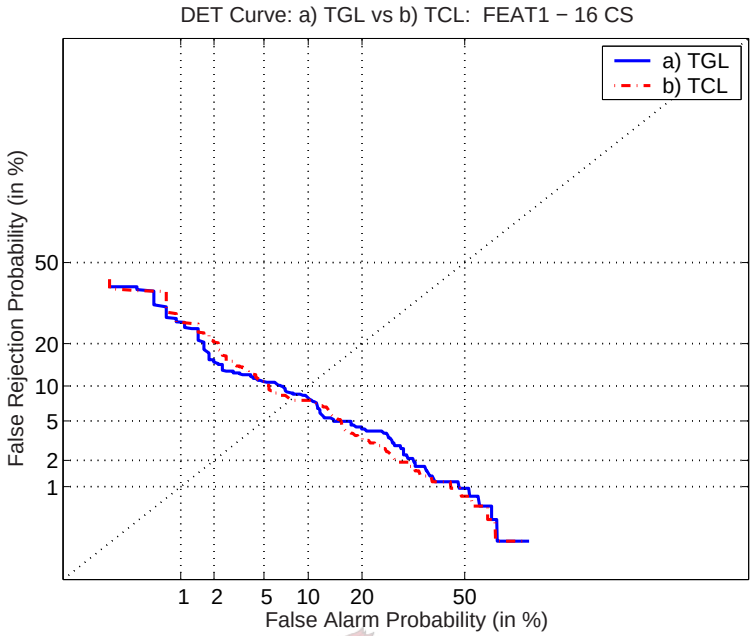


Figure B.12: *Comparison of DET curves of experiments that trained genuine labels (TGL) and experiments that trained with classified labels (TCL) using 16 conversation sides and feature set FEAT1.*

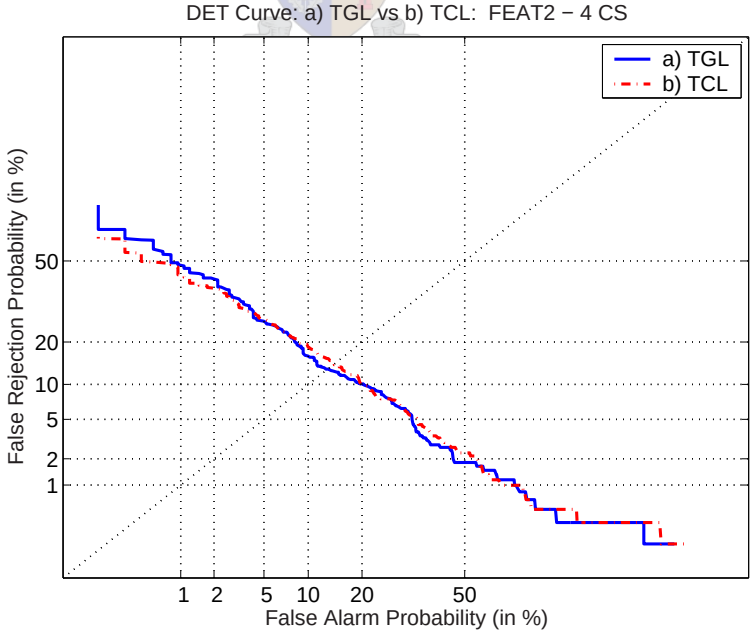


Figure B.13: *Comparison of DET curves of experiments that trained genuine labels (TGL) and experiments that trained with classified labels (TCL) using 4 conversation sides and feature set FEAT2.*

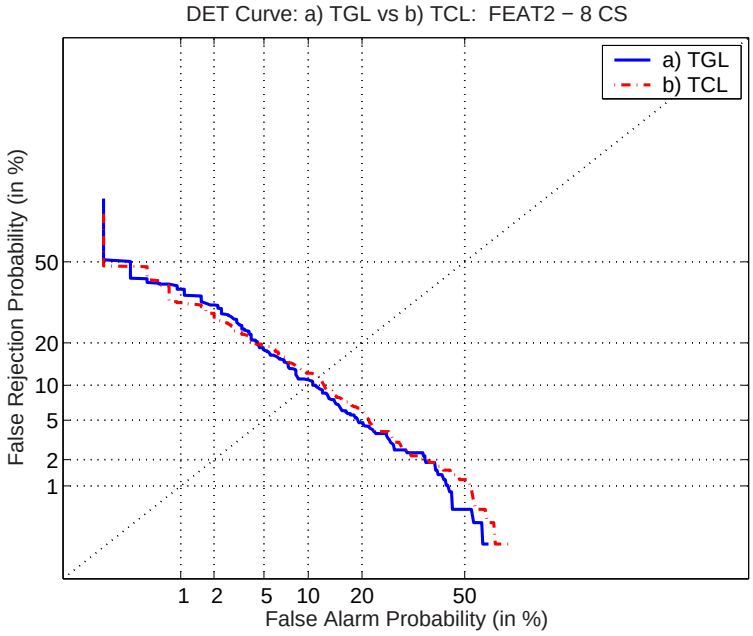


Figure B.14: *Comparison of DET curves of experiments that trained genuine labels (TGL) and experiments that trained with classified labels (TCL) using 8 conversation sides and feature set FEAT2.*

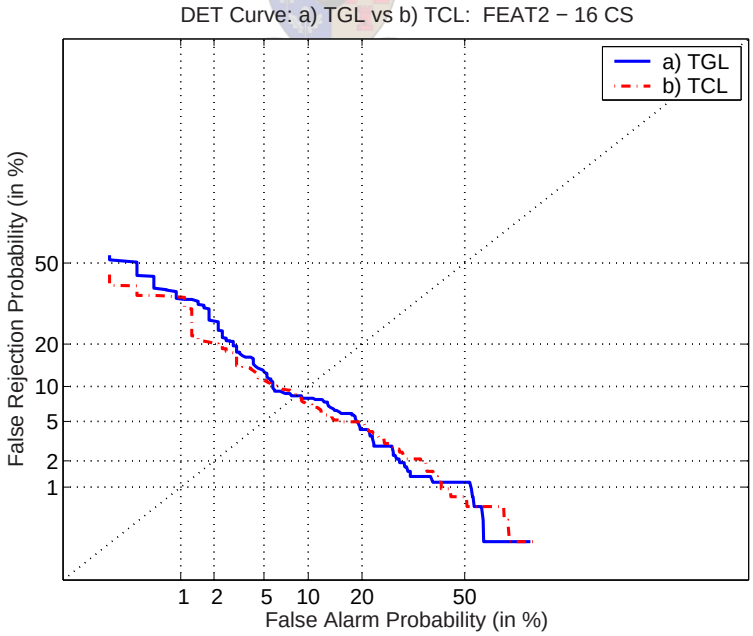


Figure B.15: *Comparison of DET curves of experiments that trained genuine labels (TGL) and experiments that trained with classified labels (TCL) using 16 conversation sides and feature set FEAT2.*

Appendix C

Verifier Combination Techniques

C.1 DET Curves

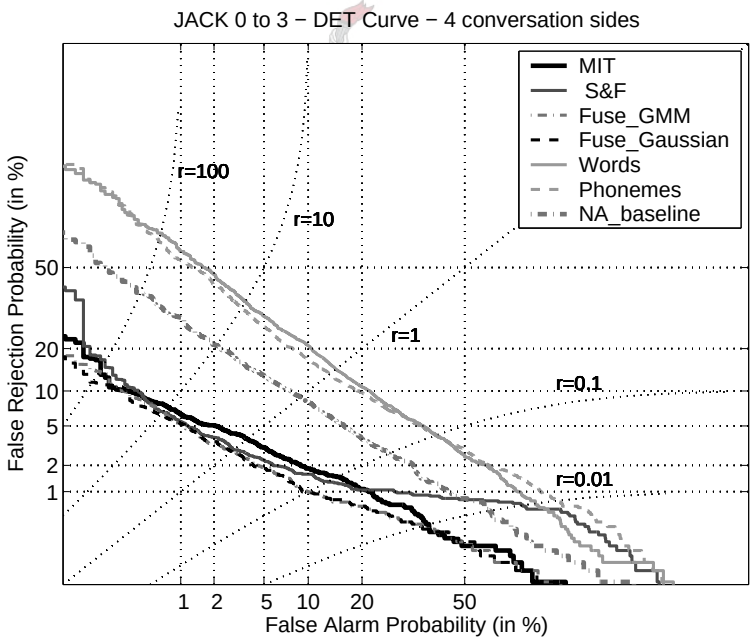


Figure C.1: *DET curves comparing the different verifier combination techniques with their constituent verifiers, using 4 conversation sides for training.*

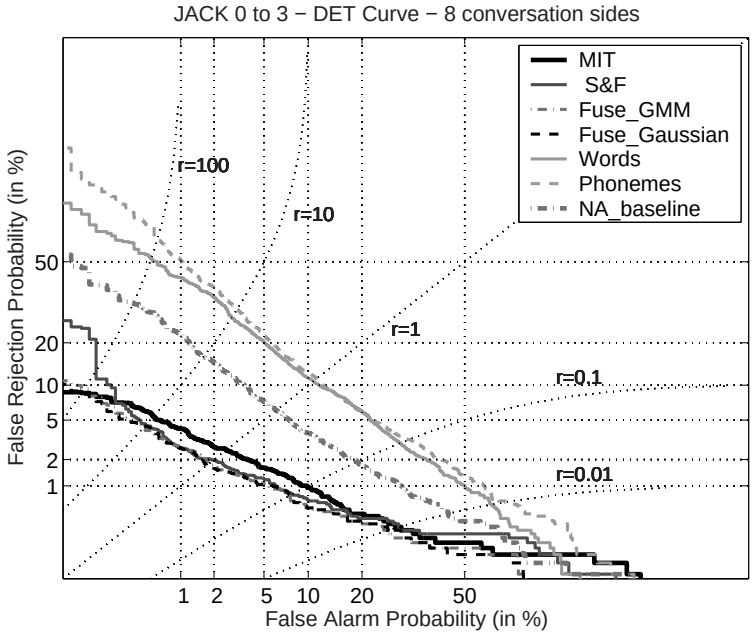


Figure C.2: *DET curves comparing the different verifier combination techniques with their constituent verifiers, using 8 conversation sides for training.*

C.2 Significant Probability levels of Fusion Techniques

The level of significant improvement of the fusion techniques relative to the MIT verifier (Verifier C) for *Fuse_Gaussian*, *Fuse_GMM* and *S&F* are shown in Figures C.3 to C.11. These are plotted against $r = FRR/FAR$. This is done by making use of the McNemar test as in Section 5.2.

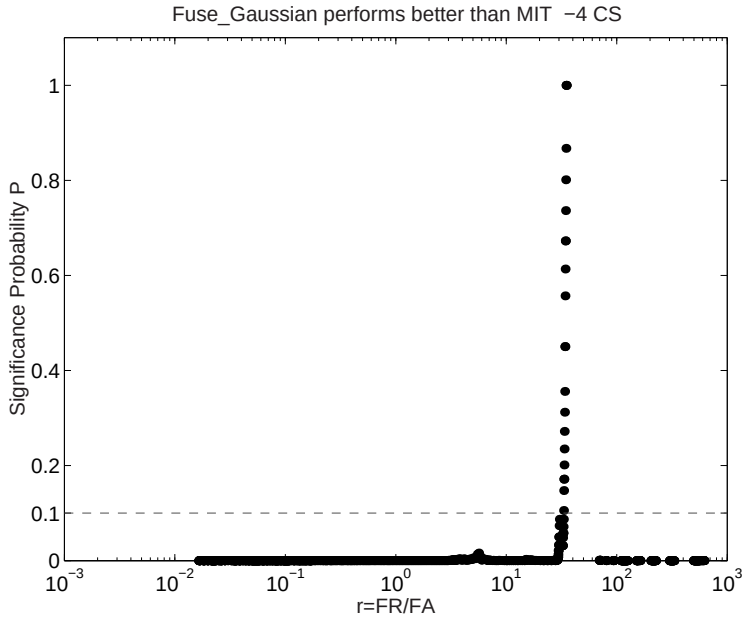


Figure C.3: *Significant Probability levels of "Fuse_Gaussian" performing better than verifier C vs the ratio of false rejection rate over false alarm rate ($r = FRR/FAR$), using 4 conversation sides (CS) for training.*

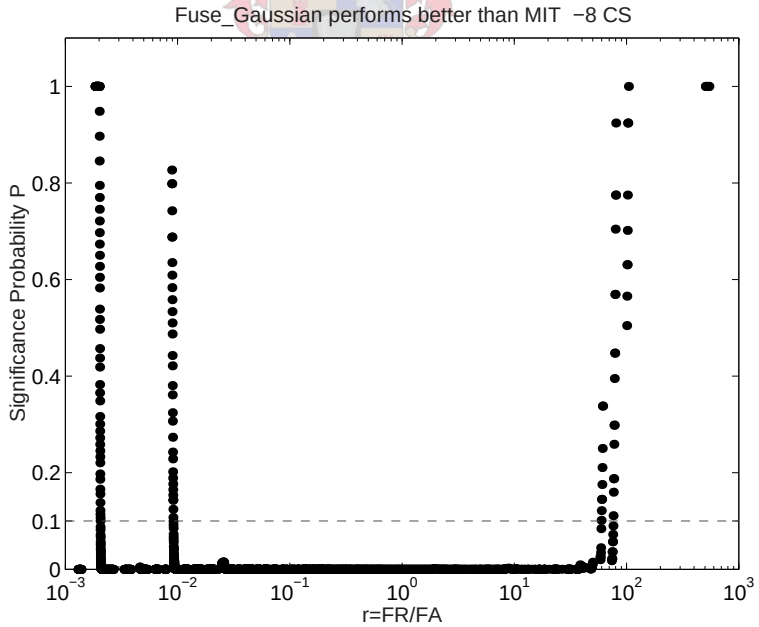


Figure C.4: *Significant Probability levels of "Fuse_Gaussian" performing better than verifier C vs the ratio of false rejection rate over false alarm rate ($r = FRR/FAR$), using 8 conversation sides (CS) for training.*

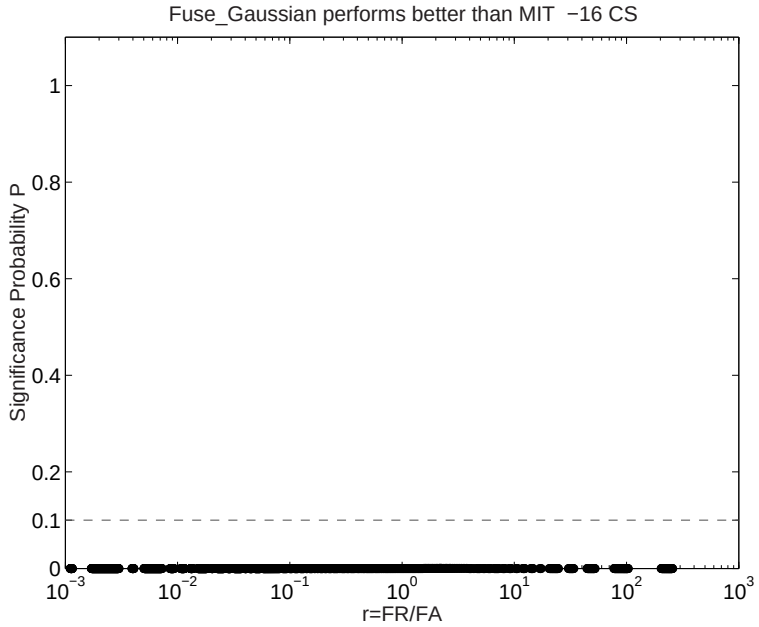


Figure C.5: *Significant Probability levels of "Fuse_Gaussian" performing better than verifier C vs the ratio of false rejection rate over false alarm rate ($r = FRR/FAR$), using 16 conversation sides (CS) for training.*

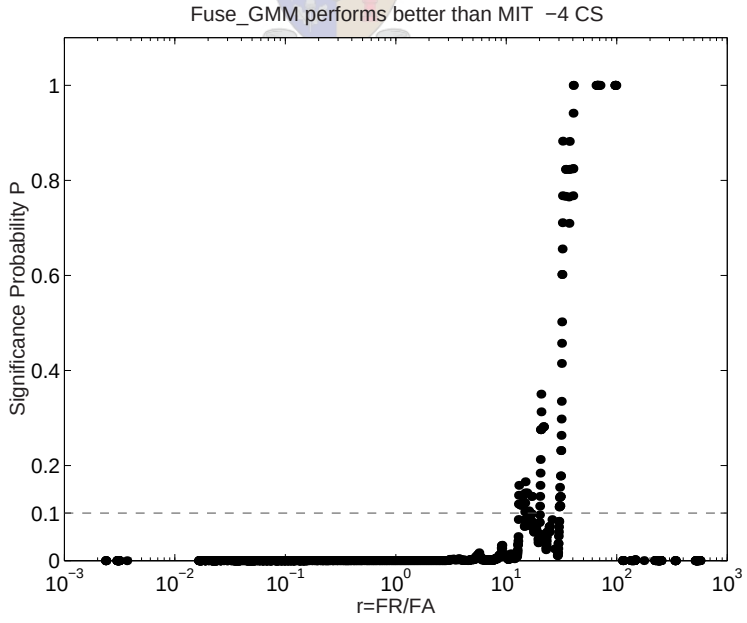


Figure C.6: *Significant Probability levels of "Fuse_GMM" performing better than verifier C vs the ratio of false rejection rate over false alarm rate ($r = FRR/FAR$), using 4 conversation sides (CS) for training.*

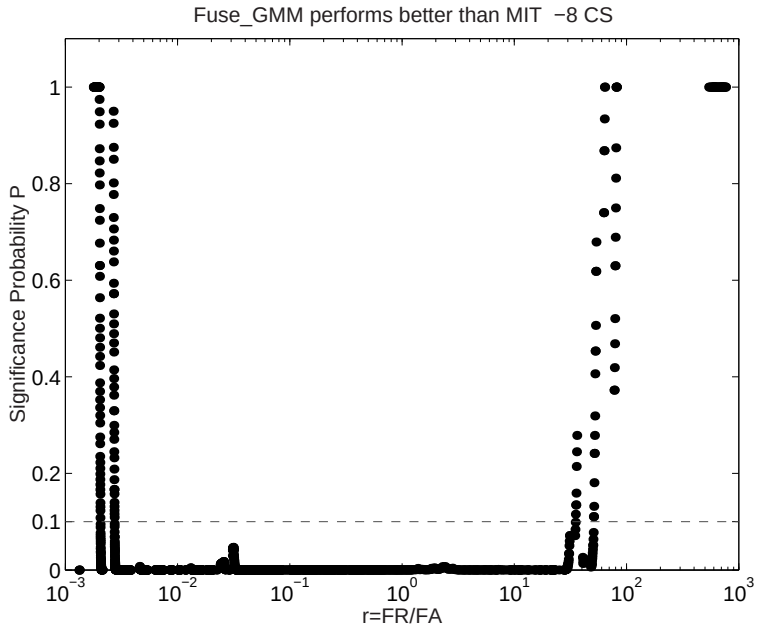


Figure C.7: *Significant Probability levels of "Fuse_GMM" performing better than verifier C vs the ratio of false rejection rate over false alarm rate ($r = FRR/FAR$), using 8 conversation sides (CS) for training.*

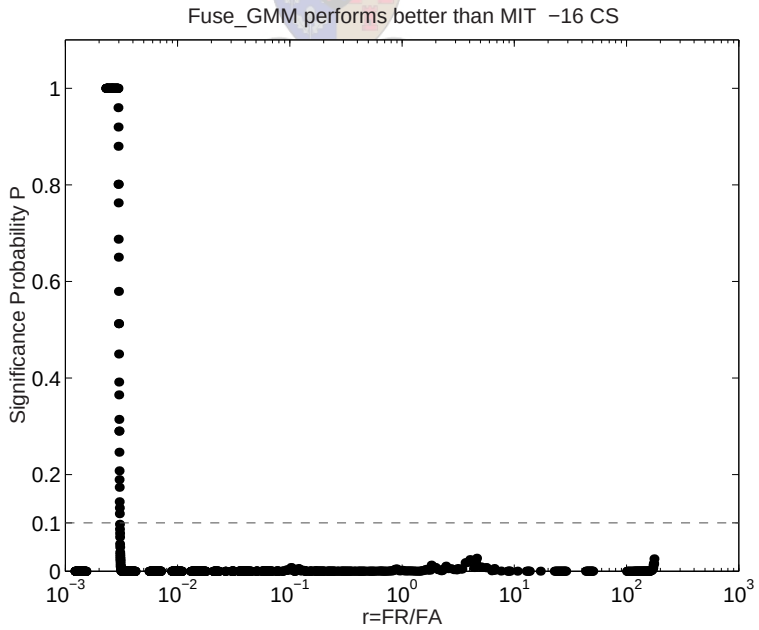


Figure C.8: *Significant Probability levels of "Fuse_GMM" performing better than verifier C vs the ratio of false rejection rate over false alarm rate ($r = FRR/FAR$), using 16 conversation sides (CS) for training.*

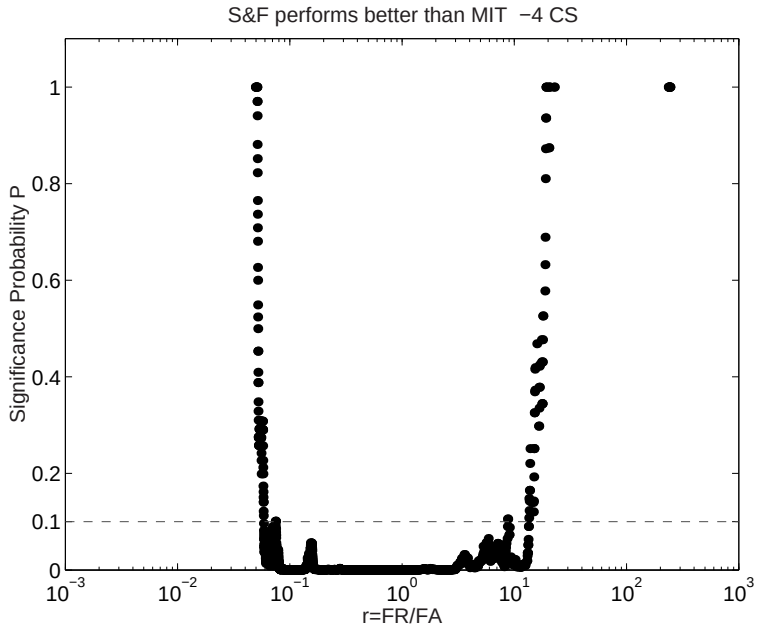


Figure C.9: *Significant Probability levels of "S&F" performing better than verifier C vs the ratio of false rejection rate over false alarm rate ($r = FRR/FAR$), using 4 conversation sides (CS) for training.*

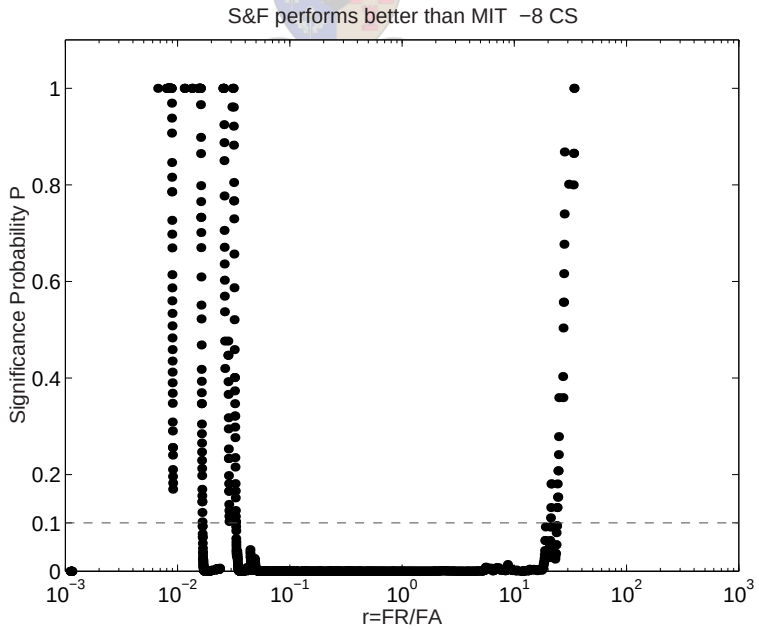


Figure C.10: *Significant Probability levels of "S&F" performing better than verifier C vs the ratio of false rejection rate over false alarm rate ($r = FRR/FAR$), using 8 conversation sides (CS) for training.*

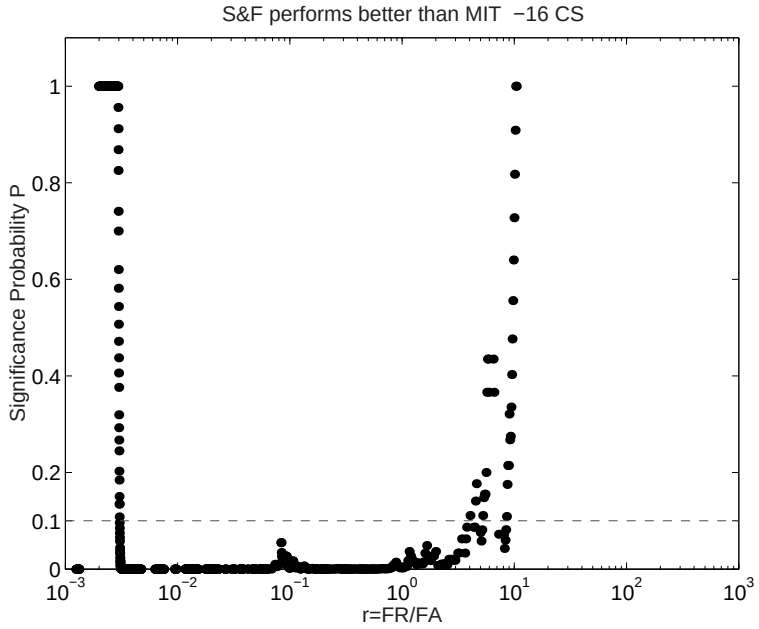


Figure C.11: *Significant Probability levels of "S&F" performing better than verifier C vs the ratio of false rejection rate over false alarm rate ($r = FRR/FAR$), using 16 conversation sides (CS) for training.*

Appendix D

NIST 2004 Evaluation

D.1 Computational statistics

Execution was divided among a set of 8 to 14 heterogeneous systems. Effective CPU clock speeds range from 2000 MHz to 3000 MHz. Some CPU's were not available for all tasks.

Feature extraction and phoneme transcriptions for system SDV_3 were performed separately from the training and trial tasks, but included files that corresponds to both 8- and 16-side training conversations.

Memory usage for feature extraction and phoneme transcription is reported as the maximum of the two tasks because they were performed sequentially.

Information for SDV_3:			
Memory usage for training:	8360	kb	
Memory usage for trails:	10000	kb	
Regarding 8-side training conversations:			
Total time for training:	2369	s	(or 40 min)
Total time for trials:	84234	s	(or 23 h 24 min)
Regarding 16-side training conversations:			
Total time for training:	2992	s	(or 52 min)
Total time for trials:	23363	s	(or 6 h 29 min)

Summary of memory usage and execution times:

Feature extraction and phoneme transcription:

Memory usage:	45500	kb	
Total time for training data:	616840	s	(or 171 h)
Total time for trial data:	226830	s	(or 63 h)

Information for SDV_2:

Memory usage for training:	8870	kb	
Memory usage for trails:	19000	kb	
Regarding 8-side training conversations:			
Total time for training:	144	s	(or 2.4 min)
Total time for trials:	2175	s	(or 36 min)
Regarding 16-side training conversations:			
Total time for training:	74	s	(or 1.2 min)
Total time for trials:	640	s	(or 11 min)

D.2 DET Curves

D.2.1 8 Conversation Sides - English Language Single Handset (ELSH)

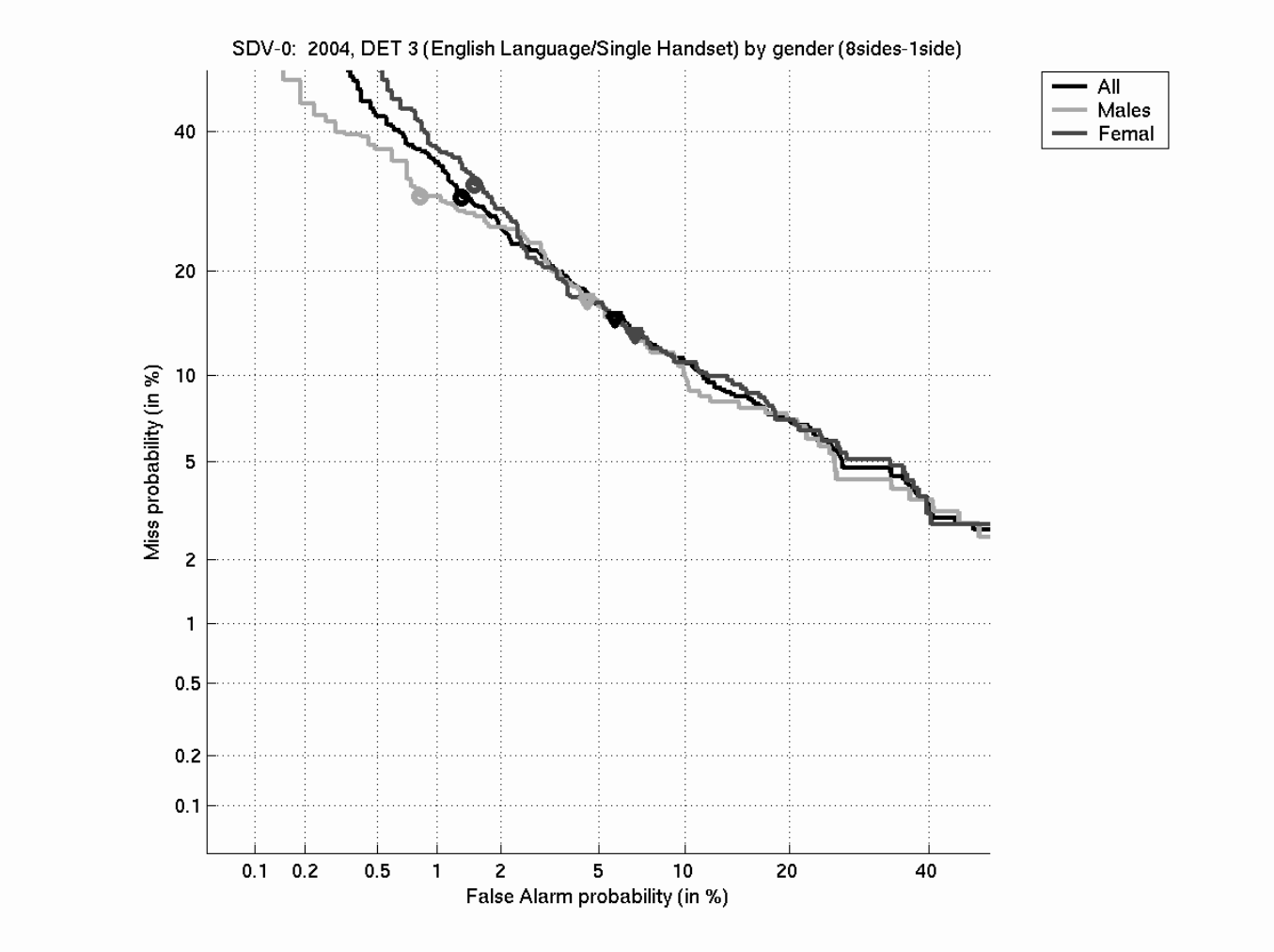


Figure D.1: *DET curves (pooled gender, male and female trials) of SDV_0 (ELSH data), using 8 (5 minute) conversation sides for training and 1 conversation side for test segment trials.*

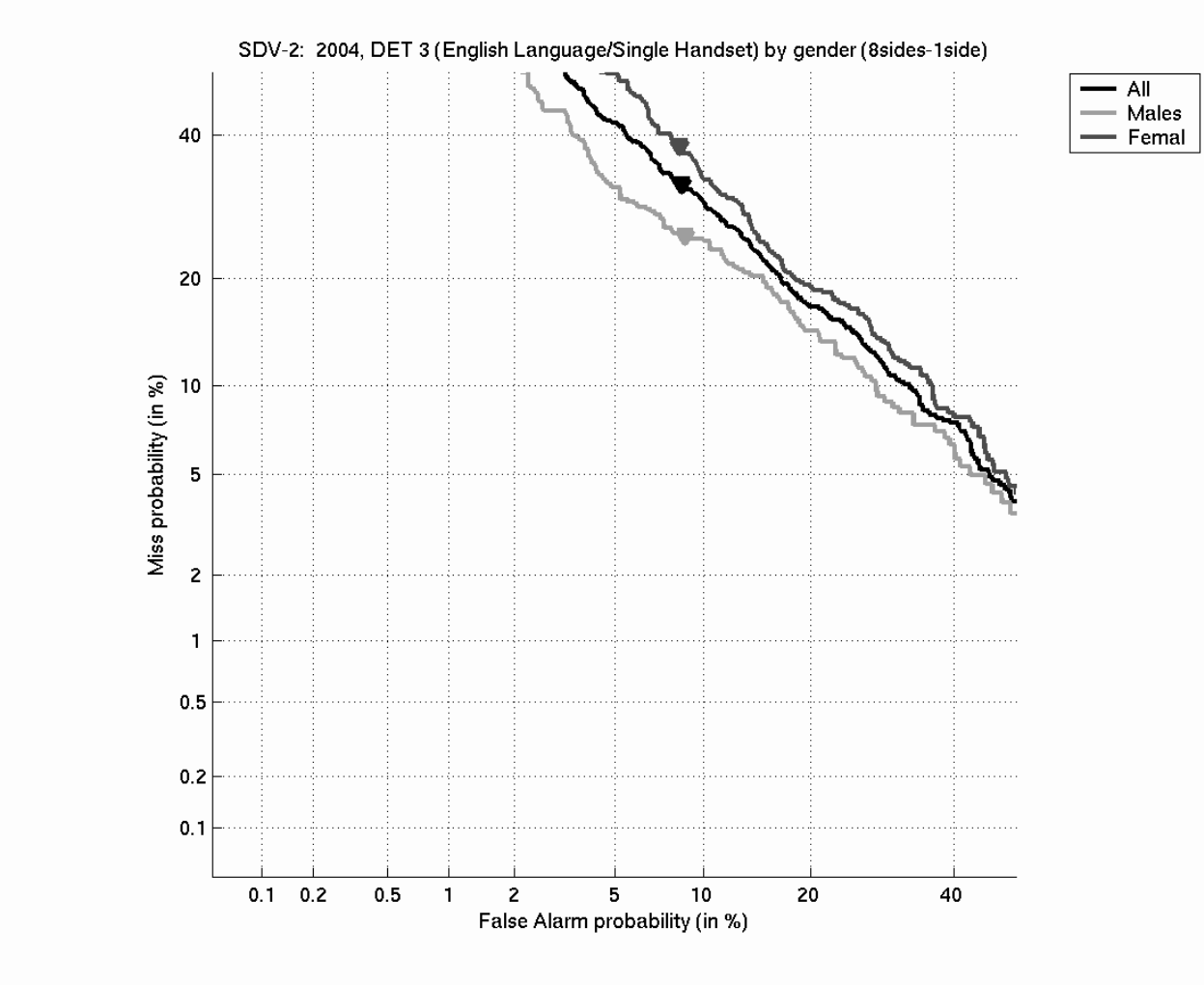


Figure D.2: *DET curves (pooled gender, male and female trials) of SDV_2 (ELSH data), using 8 (5 minute) conversation sides for training and 1 conversation side for test segment trials.*

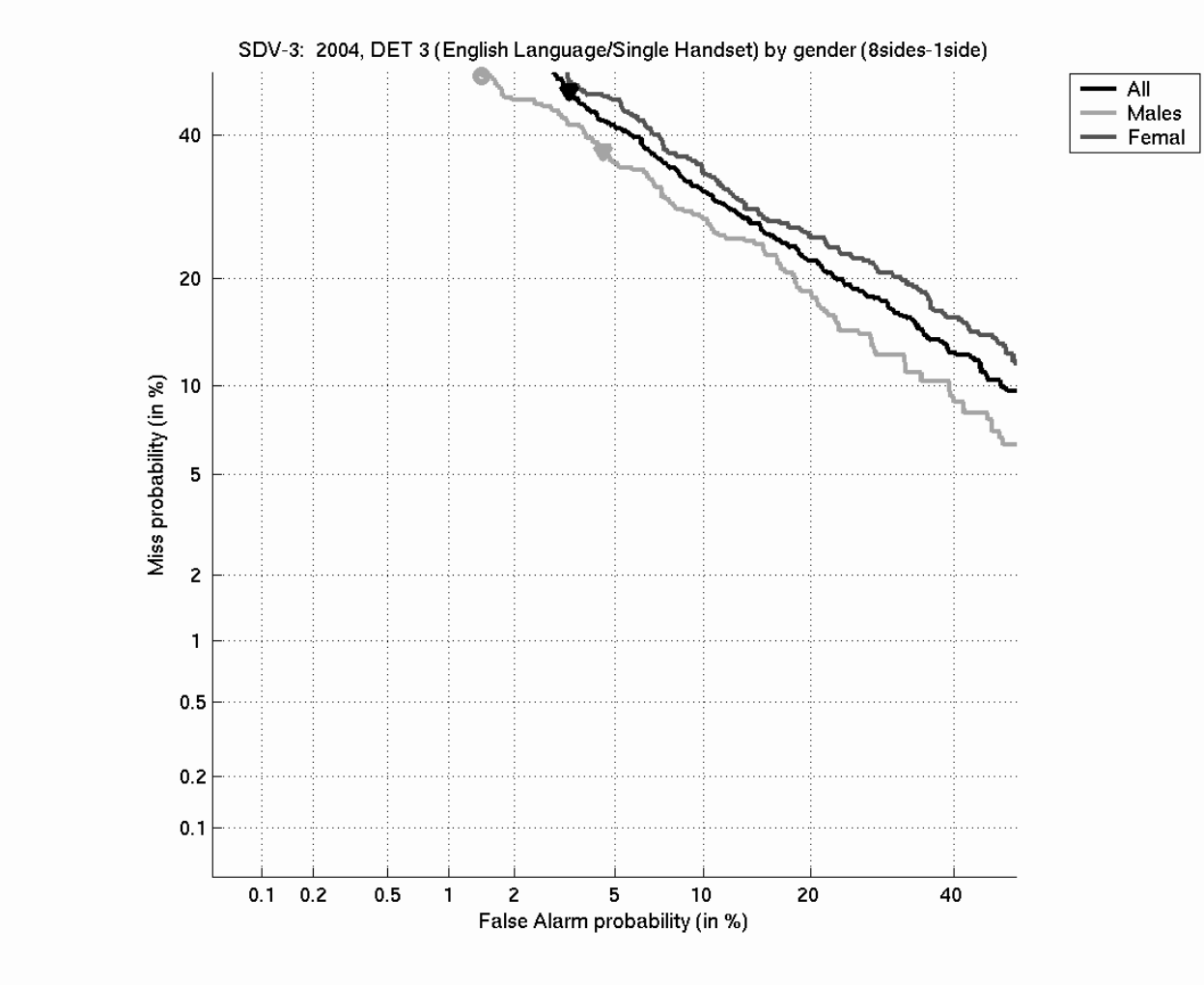


Figure D.3: *DET curves (pooled gender, male and female trials) of SDV_3 (ELSH data), using 8 (5 minute) conversation sides for training and 1 conversation side for test segment trials.*

D.2.2 16 Conversation Sides - English Language Single Handset

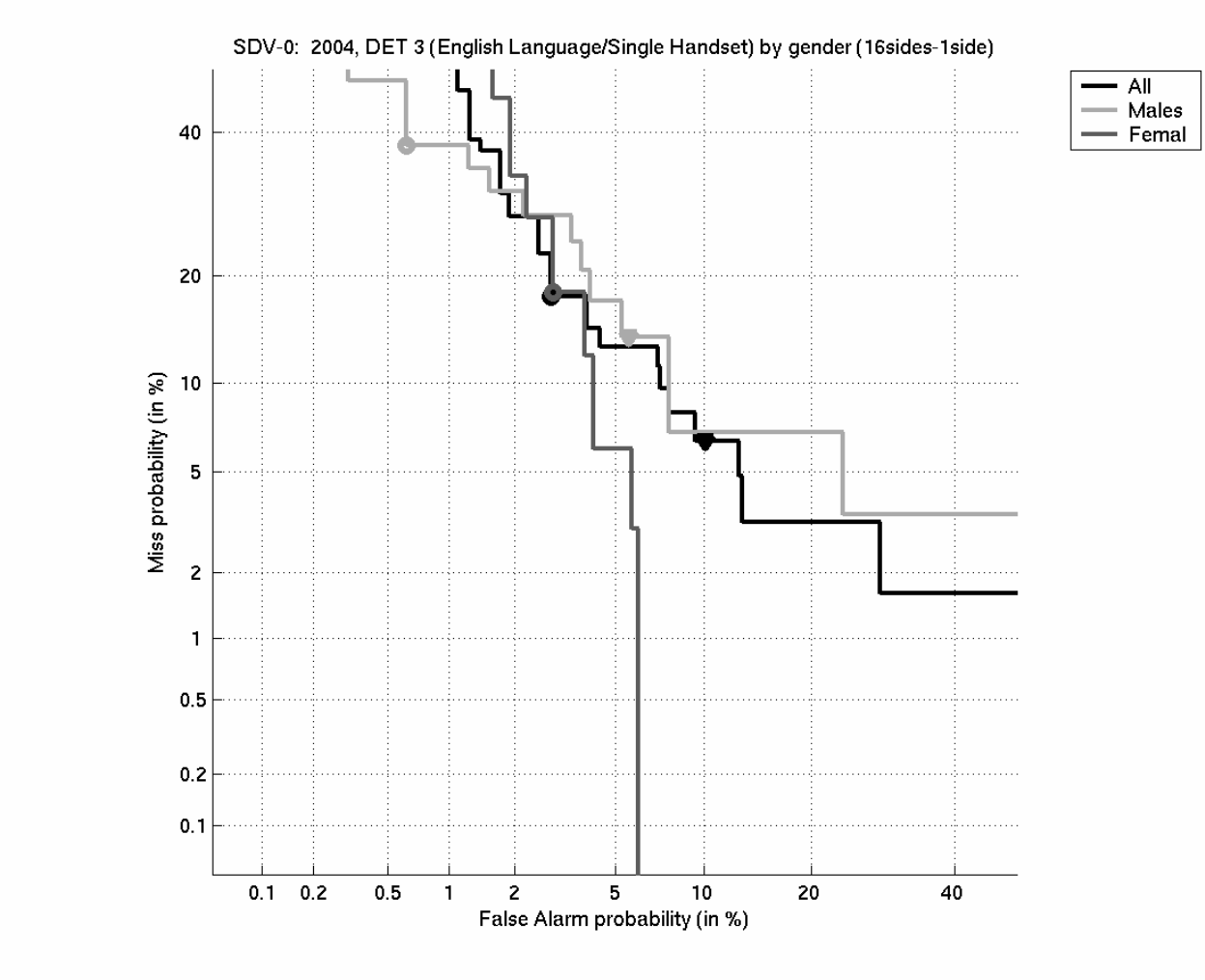


Figure D.4: *DET curves (pooled gender, male and female trials) of SDV_0 (ELSH data), using 16 (5 minute) conversation sides for training and 1 conversation side for test segment trials.*

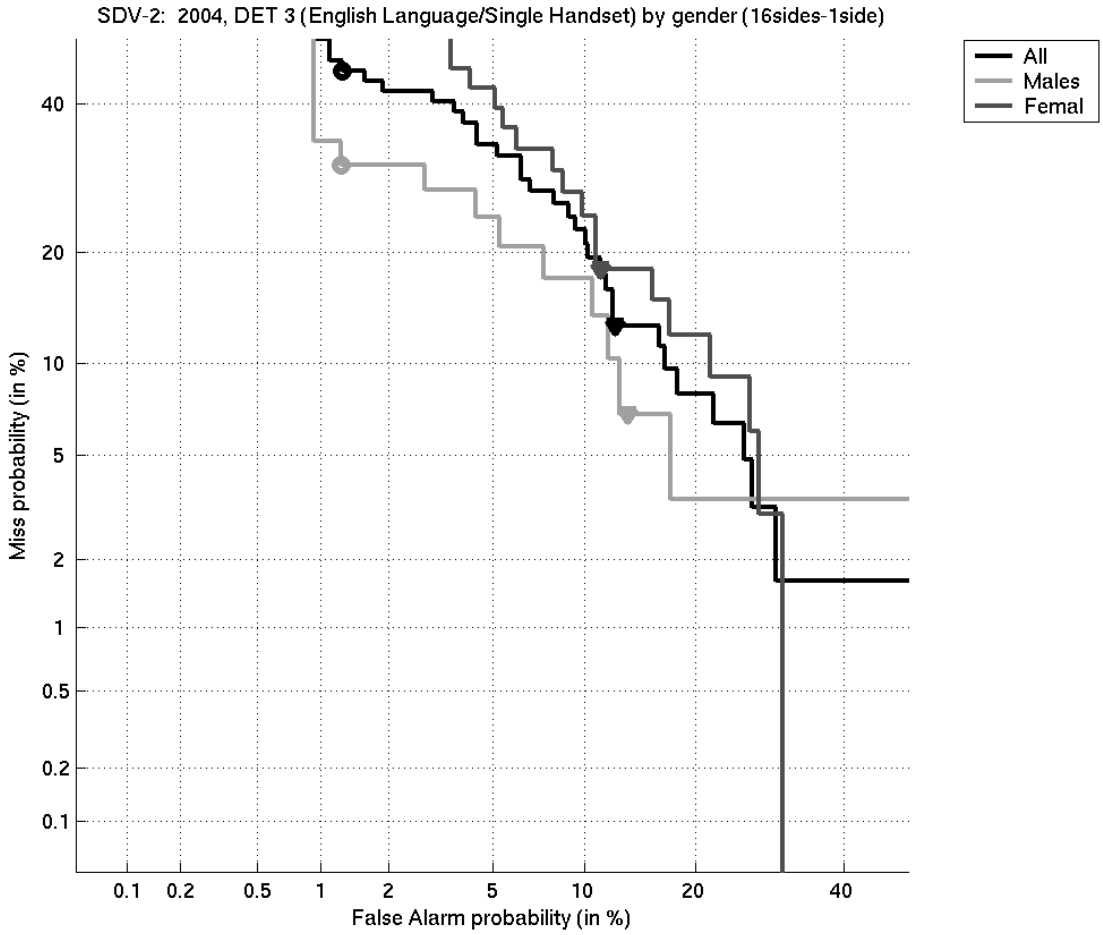


Figure D.5: *DET curves (pooled gender, male and female trials) of SDV_2 (ELSH data), using 16 (5 minute) conversation sides for training and 1 conversation side for test segment trials.*

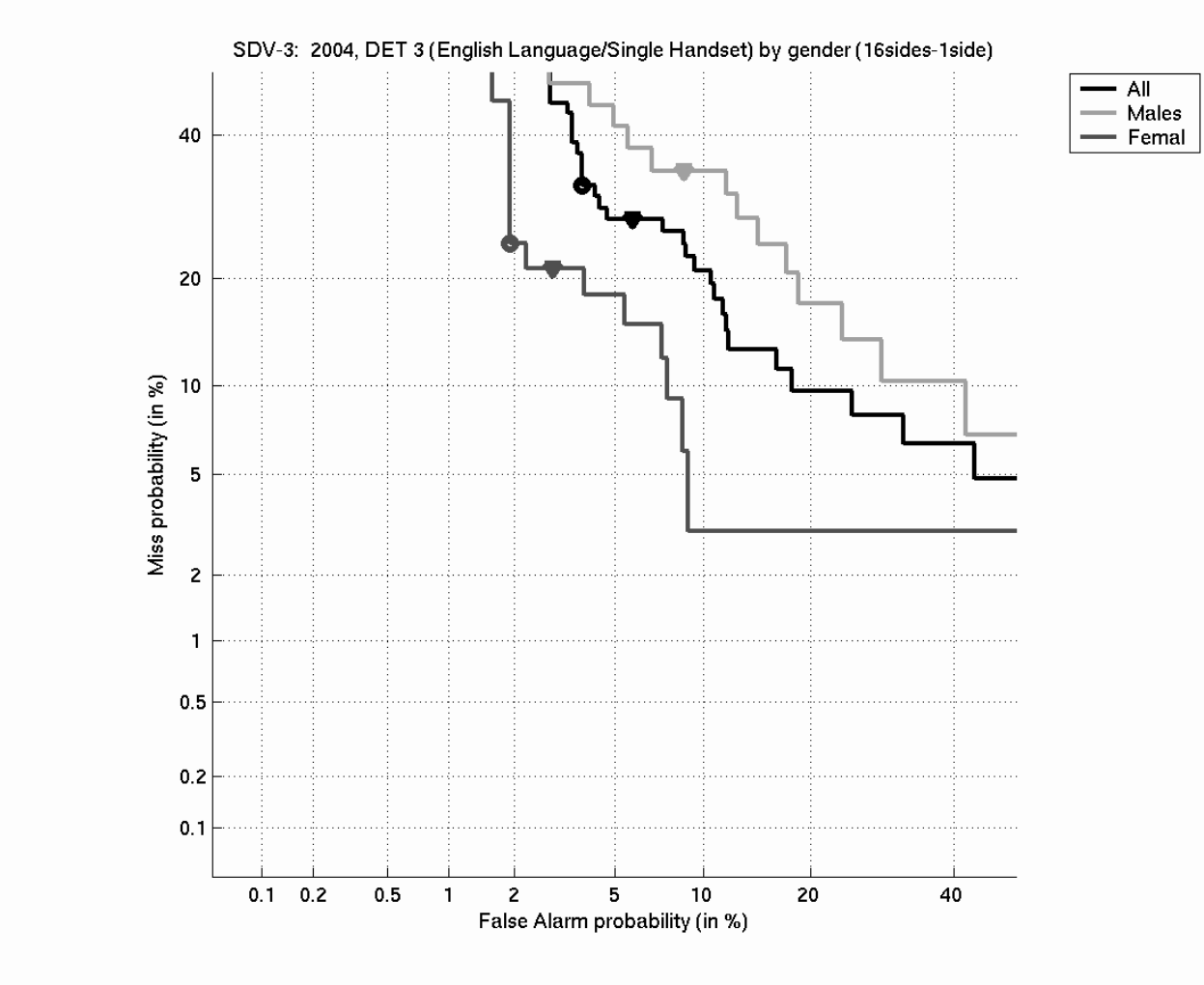


Figure D.6: *DET curves (pooled gender, male and female trials) of SDV_3 (ELSH data), using 16 (5 minute) conversation sides for training and 1 conversation side for test segment trials.*

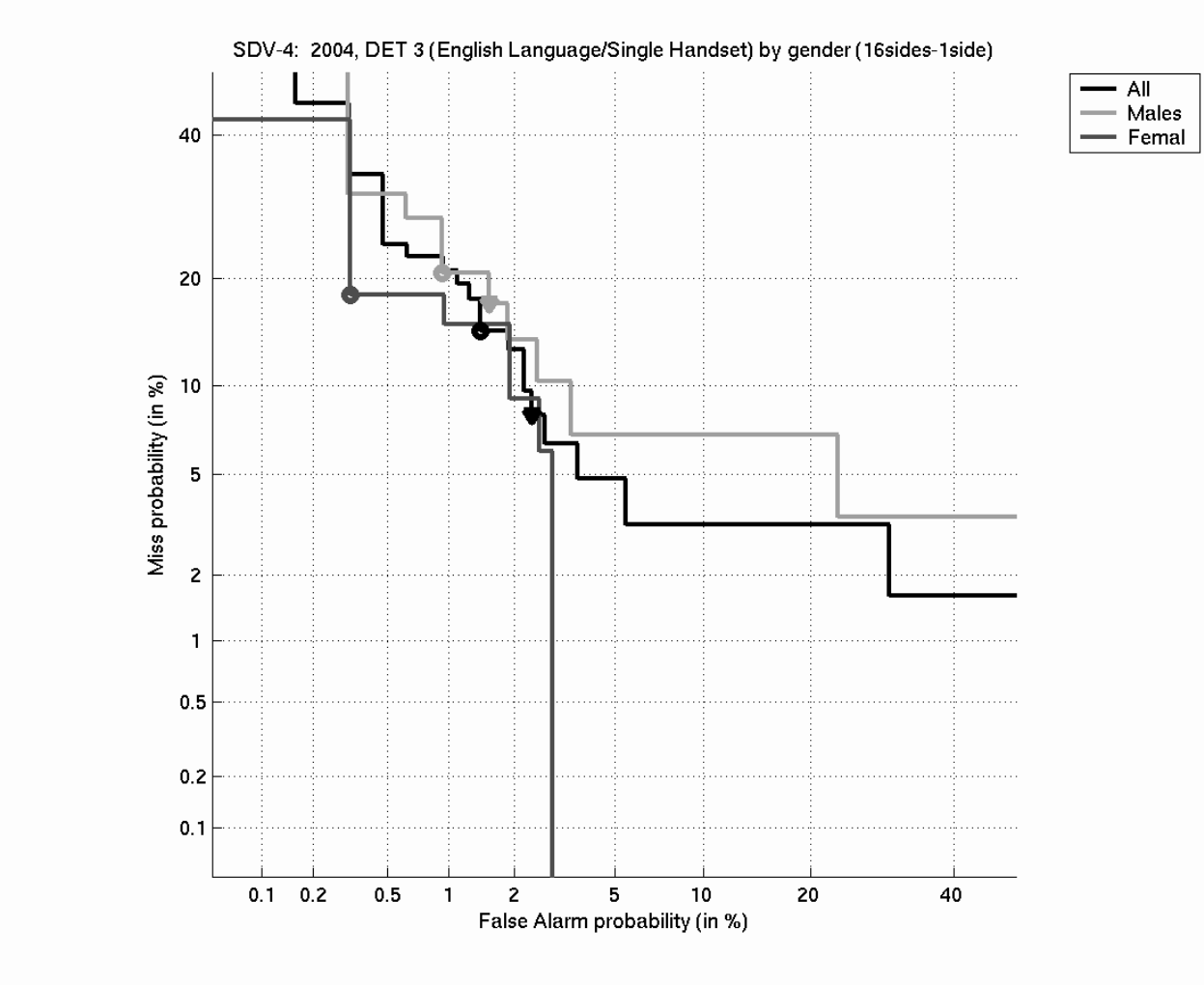


Figure D.7: *DET curves (pooled gender, male and female trials) of SDV_4 (ELSH data), using 16 (5 minute) conversation sides for training and 1 conversation side for test segment trials.*

D.2.3 8 Conversation Sides - Same and Different Language Target Trials

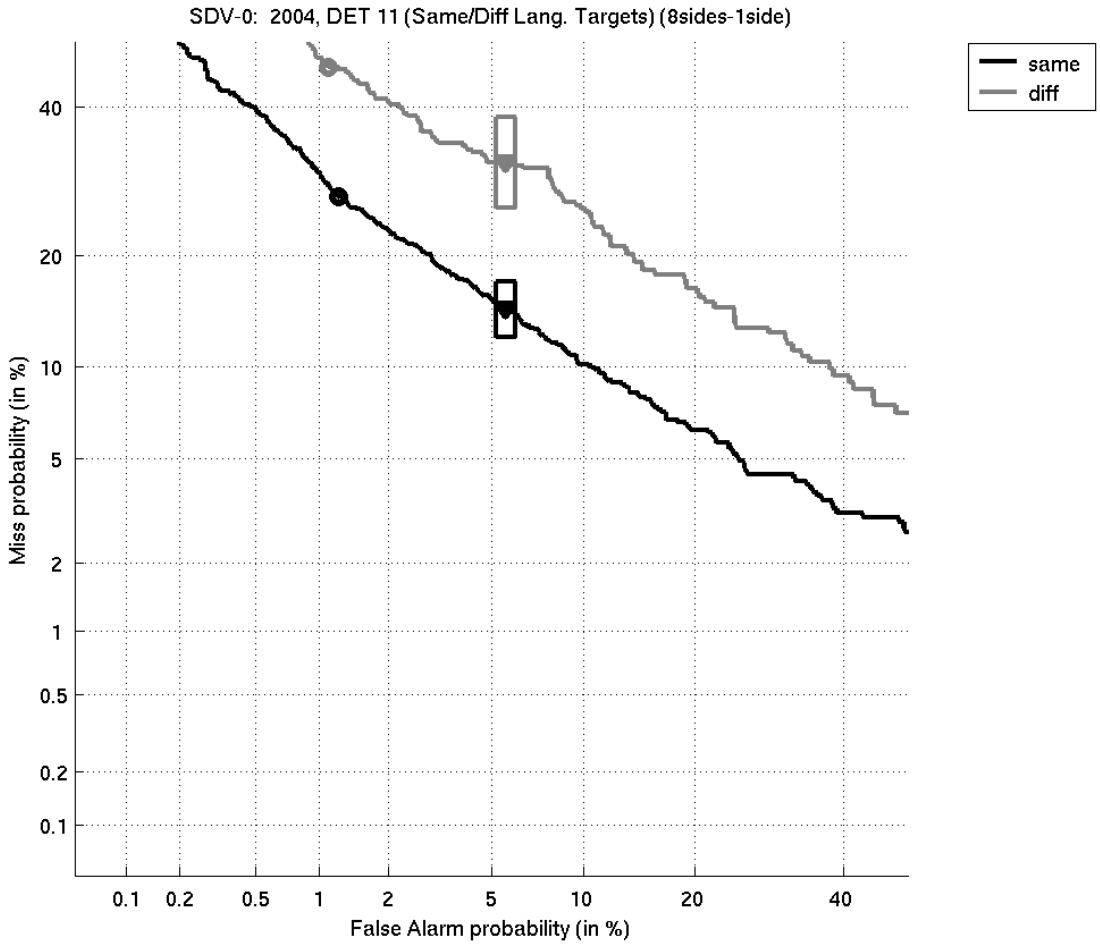


Figure D.8: *DET curves of same and different language target trials of SDV_0, using 8 (5 minute) conversation sides for training and 1 conversation side for test segment trials.*

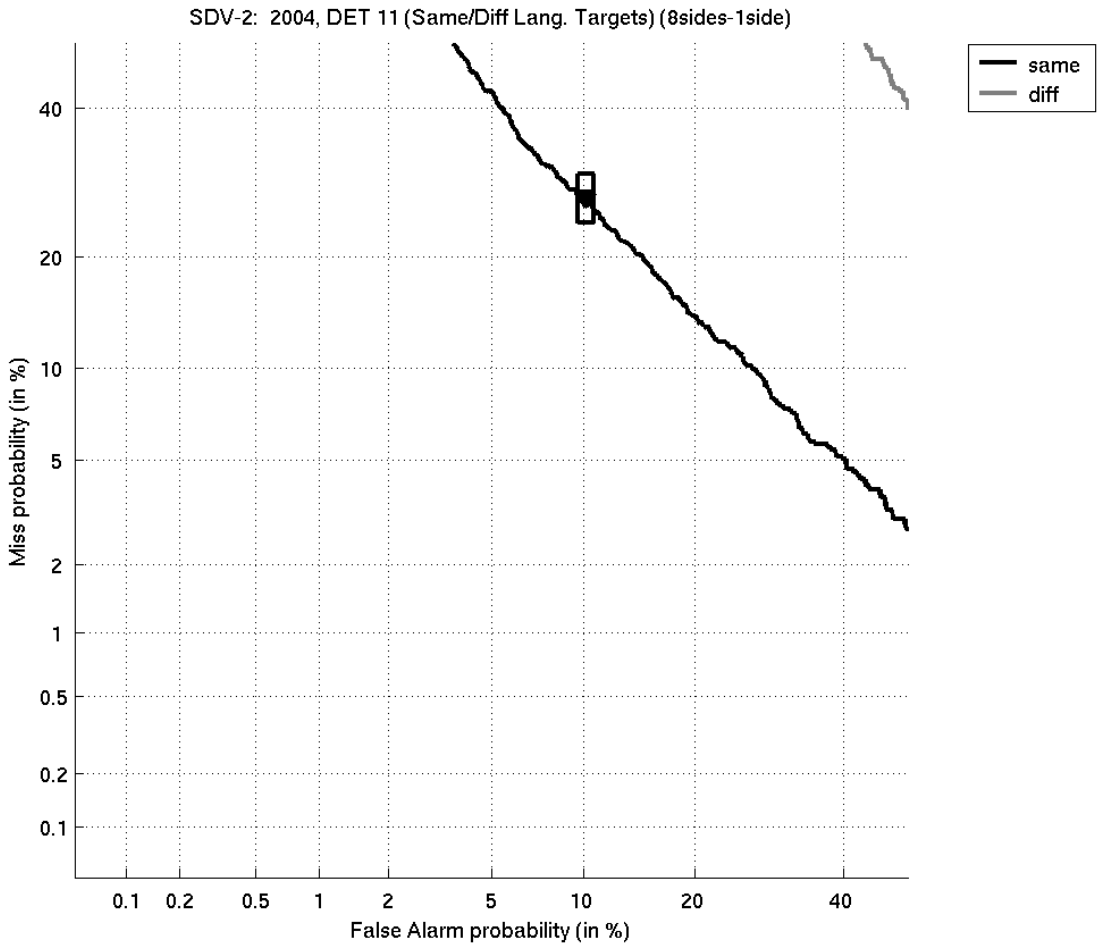


Figure D.9: *DET curves of same and different language target trials of SDV_2, using 8 (5 minute) conversation sides for training and 1 conversation side for test segment trials.*

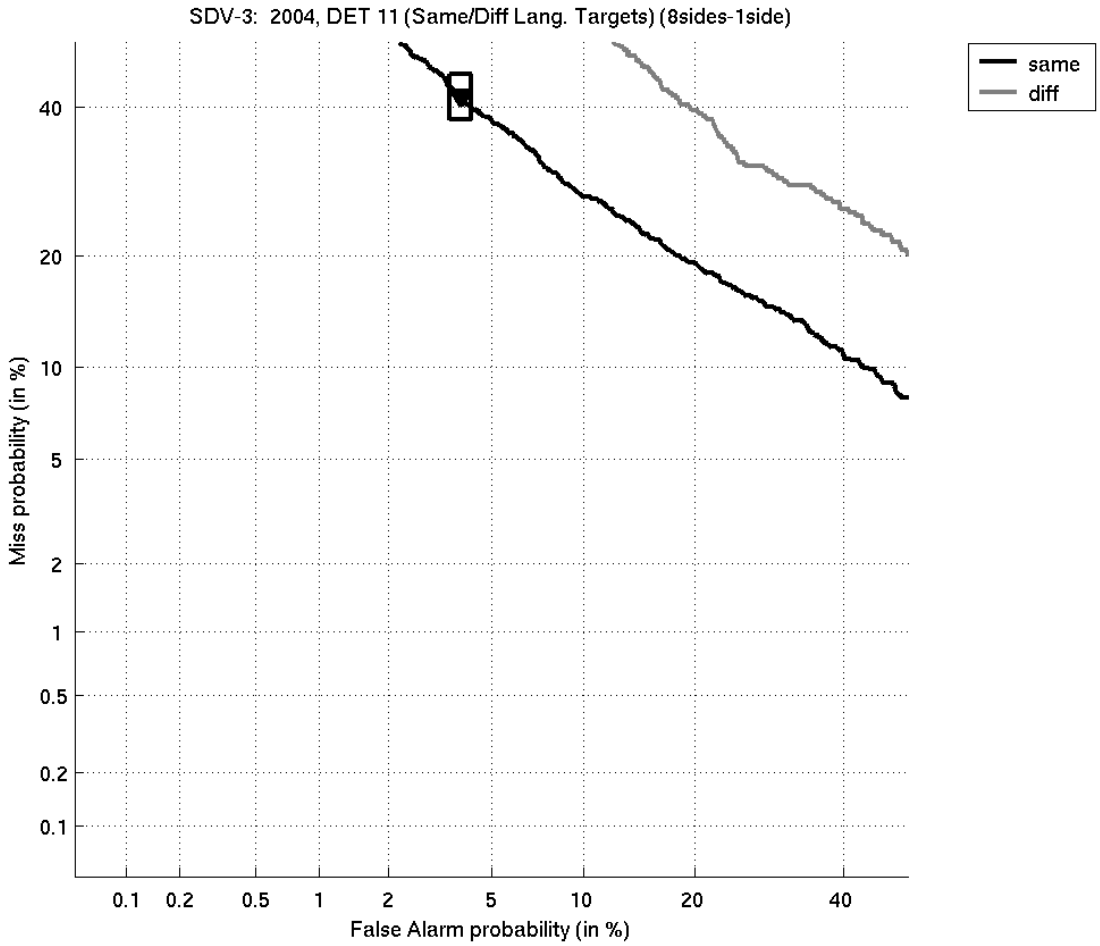


Figure D.10: *DET curves of same and different language target trials of SDV_3, using 8 (5 minute) conversation sides for training and 1 conversation side for test segment trials.*

D.2.4 8 Conversation Sides - Same and Different Language Non-Target Trials

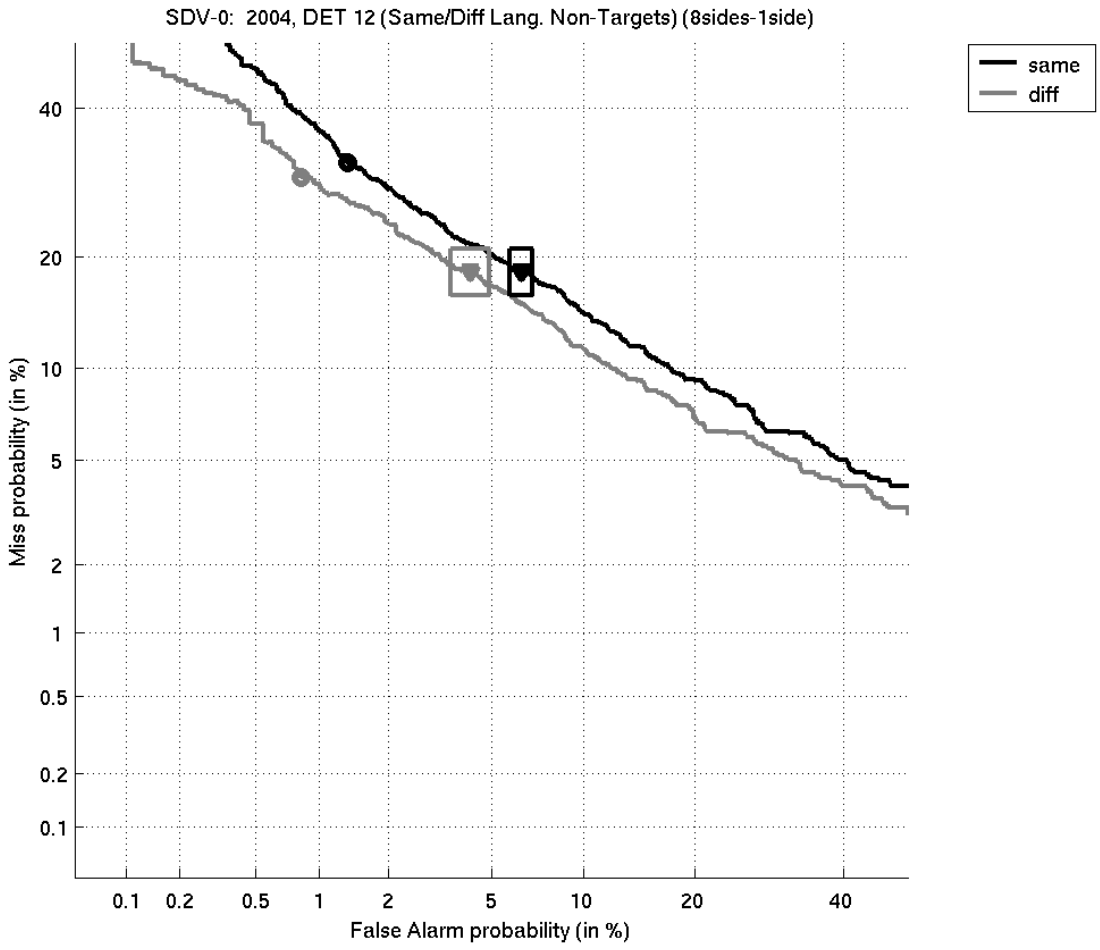


Figure D.11: *DET curves of same and different language non-target trials of SDV_0, using 8 (5 minute) conversation sides for training and 1 conversation side for test segment trials.*

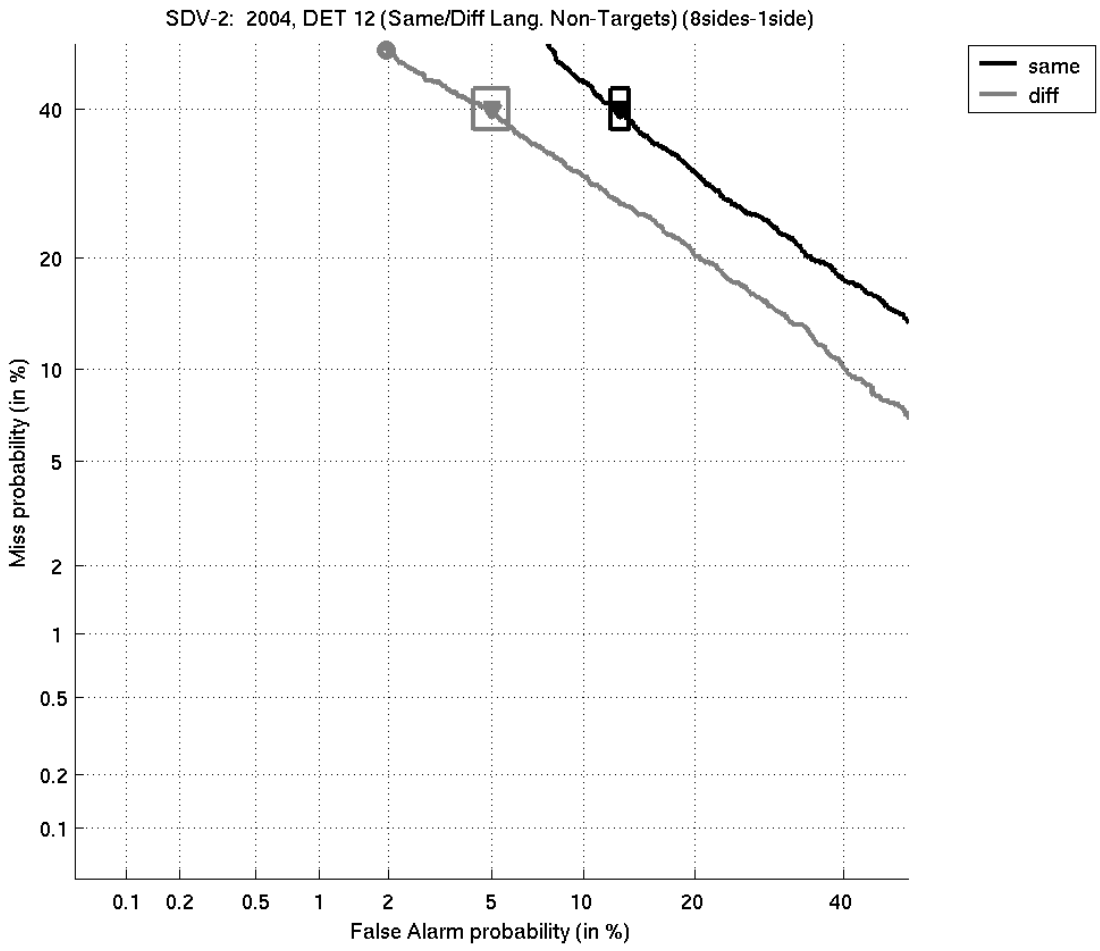


Figure D.12: *DET curves of same and different language non-target trials of SDV_2, using 8 (5 minute) conversation sides for training and 1 conversation side for test segment trials.*

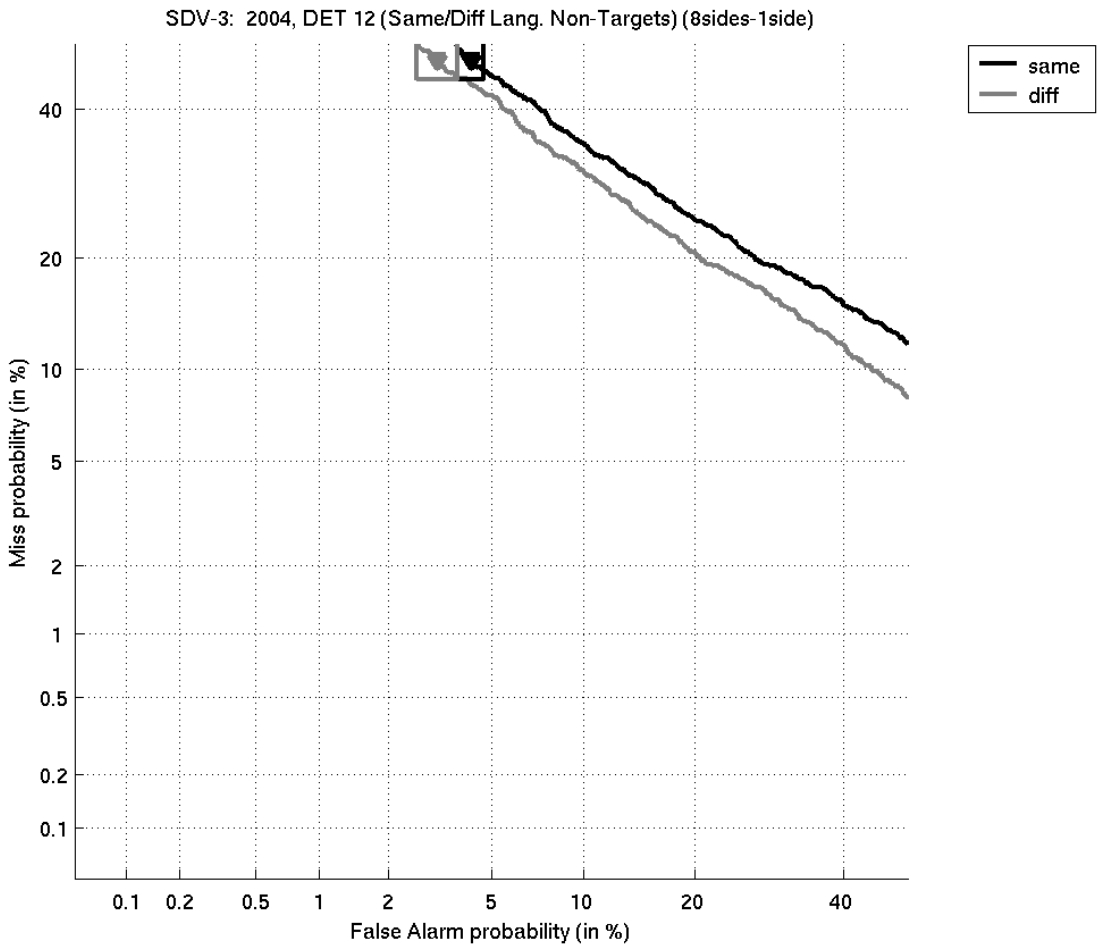


Figure D.13: *DET curves of same and different language non-target trials of SDV_3, using 8 (5 minute) conversation sides for training and 1 conversation side for test segment trials.*

D.2.5 16 Conversation Sides - Same and Different Language Target Trials

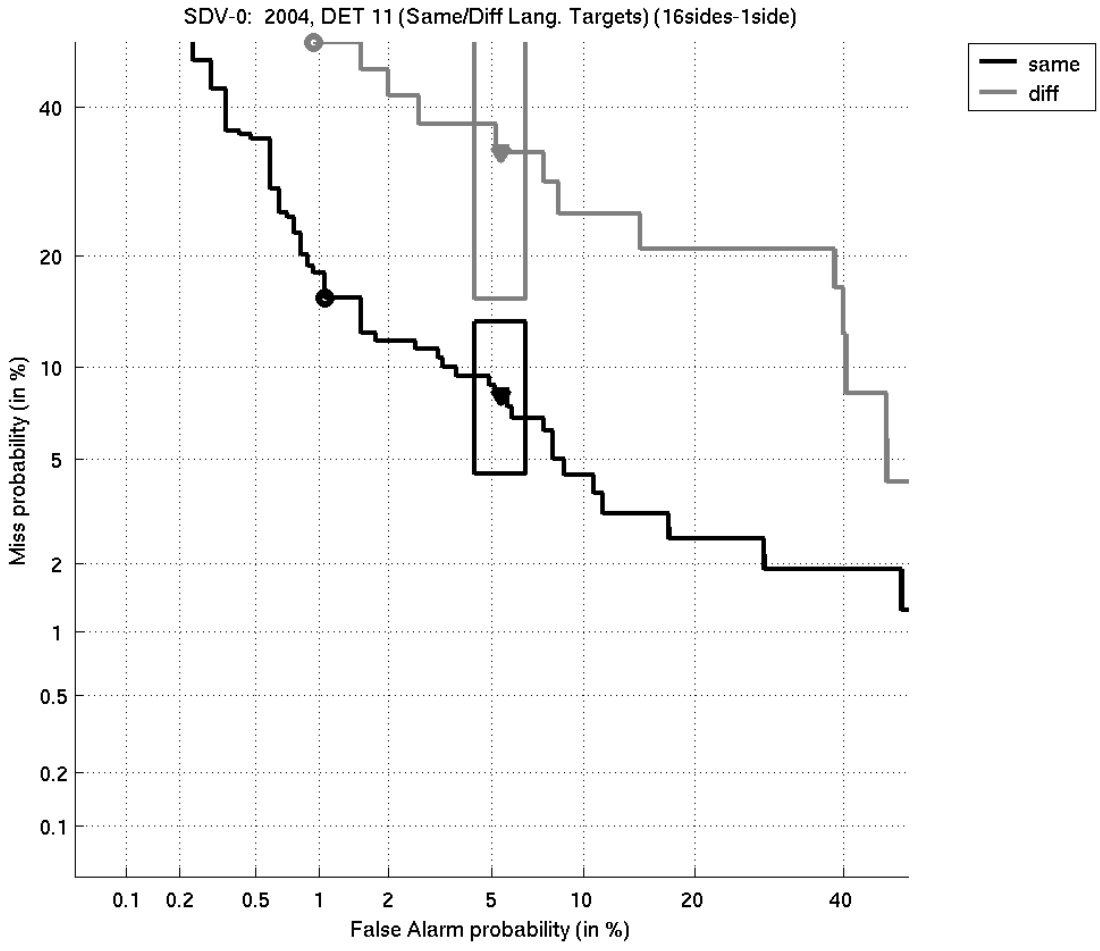


Figure D.14: *DET curves of same and different language target trials of SDV_0, using 16 (5 minute) conversation sides for training and 1 conversation side for test segment trials.*

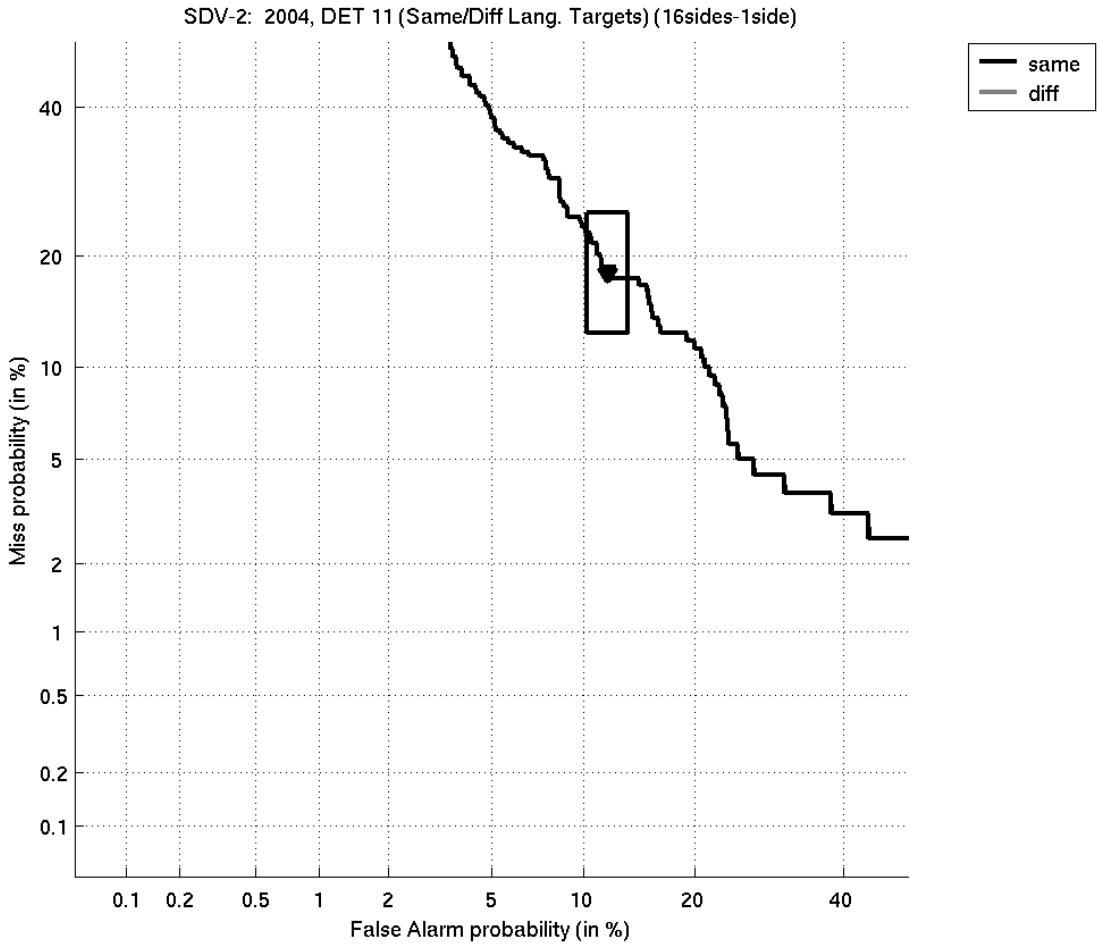


Figure D.15: *DET curves of same and different language target trials of SDV_2, using 16 (5 minute) conversation sides for training and 1 conversation side for test segment trials.*

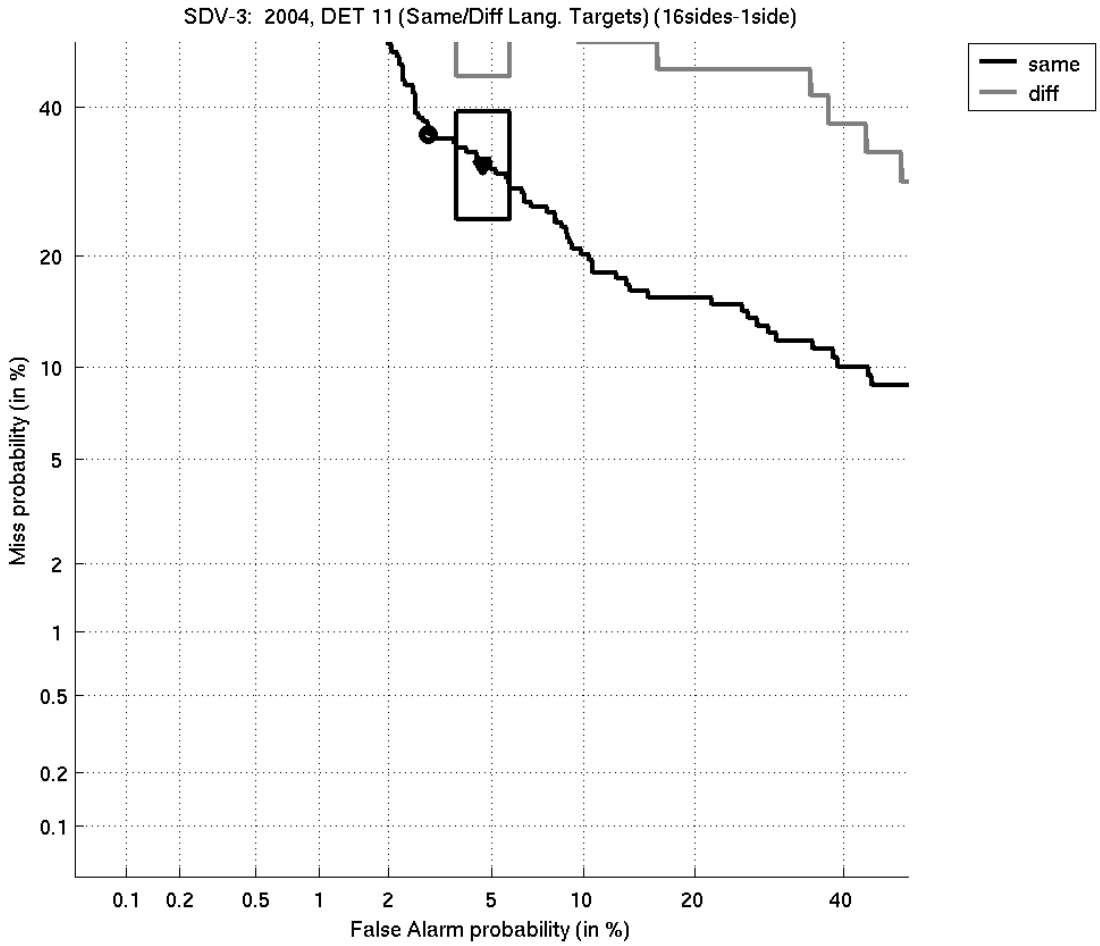


Figure D.16: *DET curves of same and different language target trials of SDV_3, using 16 (5 minute) conversation sides for training and 1 conversation side for test segment trials.*

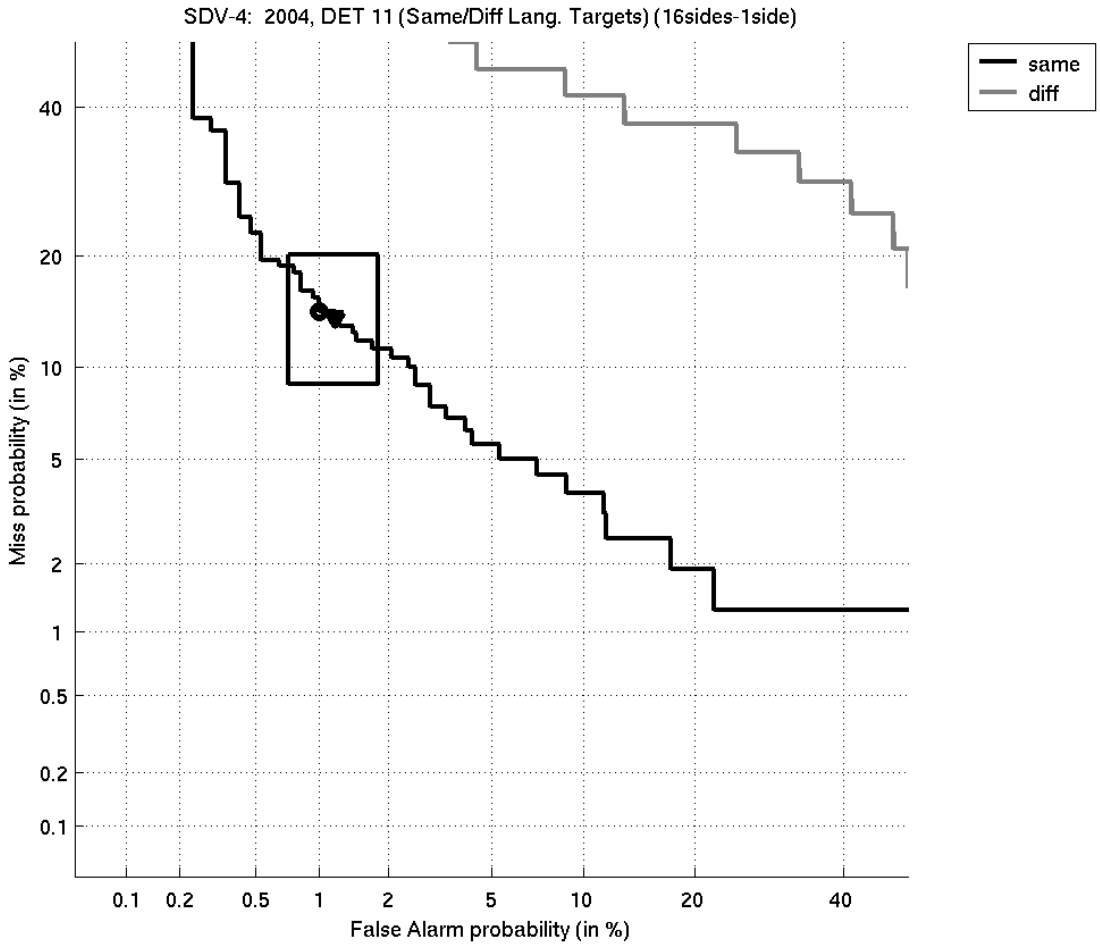


Figure D.17: *DET curves of same and different language target trials of SDV_4, using 16 (5 minute) conversation sides for training and 1 conversation side for test segment trials.*

D.2.6 16 Conversation Sides - Same and Different Language Non-Target Trials

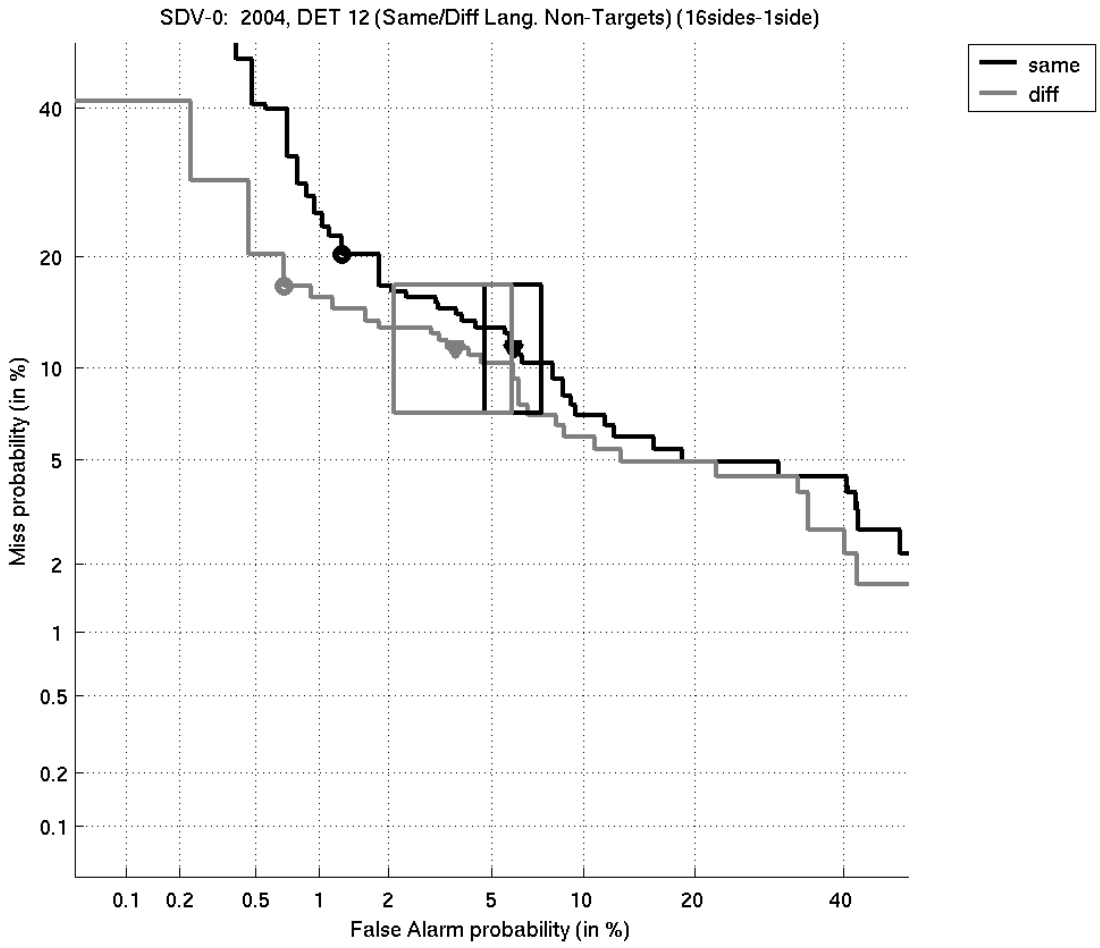


Figure D.18: *DET curves of same and different language non-target trials of SDV_0, using 16 (5 minute) conversation sides for training and 1 conversation side for test segment trials.*

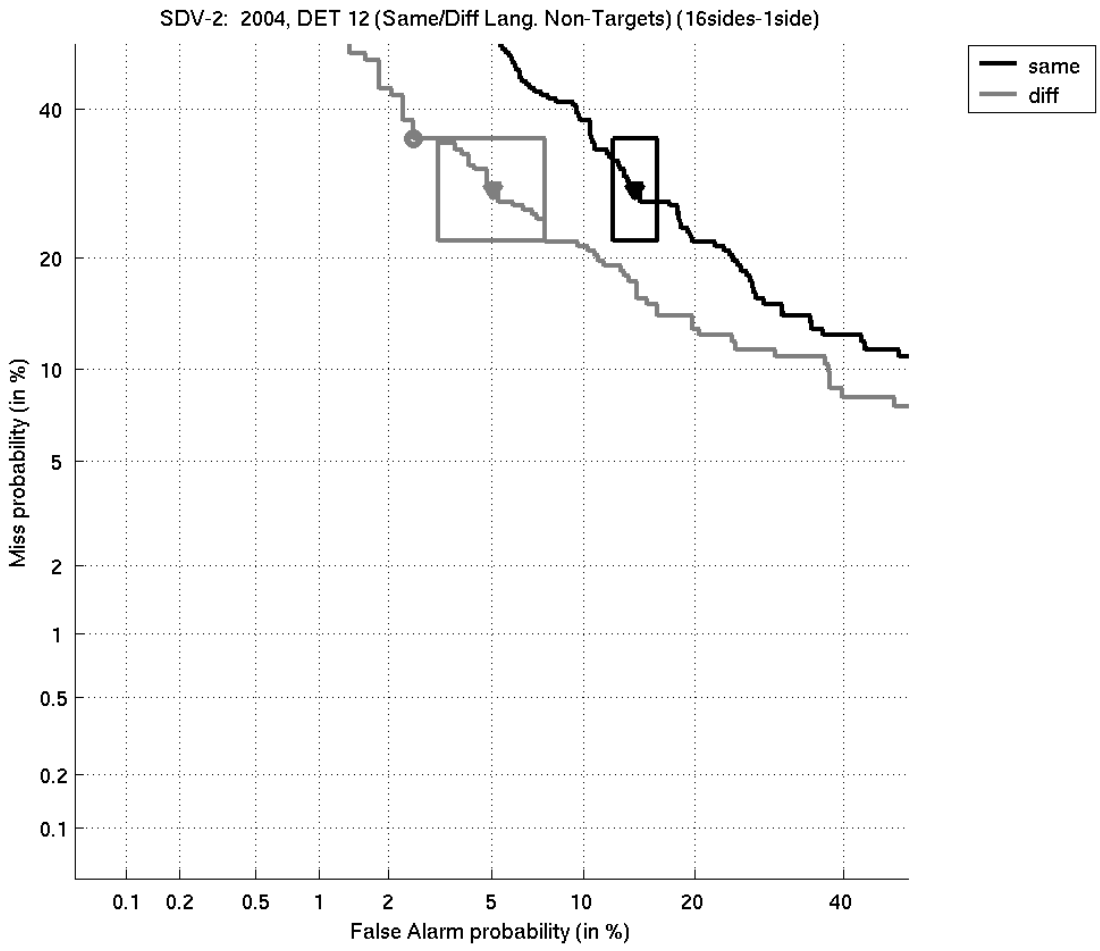


Figure D.19: *DET curves of same and different language non-target trials of SDV_2, using 16 (5 minute) conversation sides for training and 1 conversation side for test segment trials.*

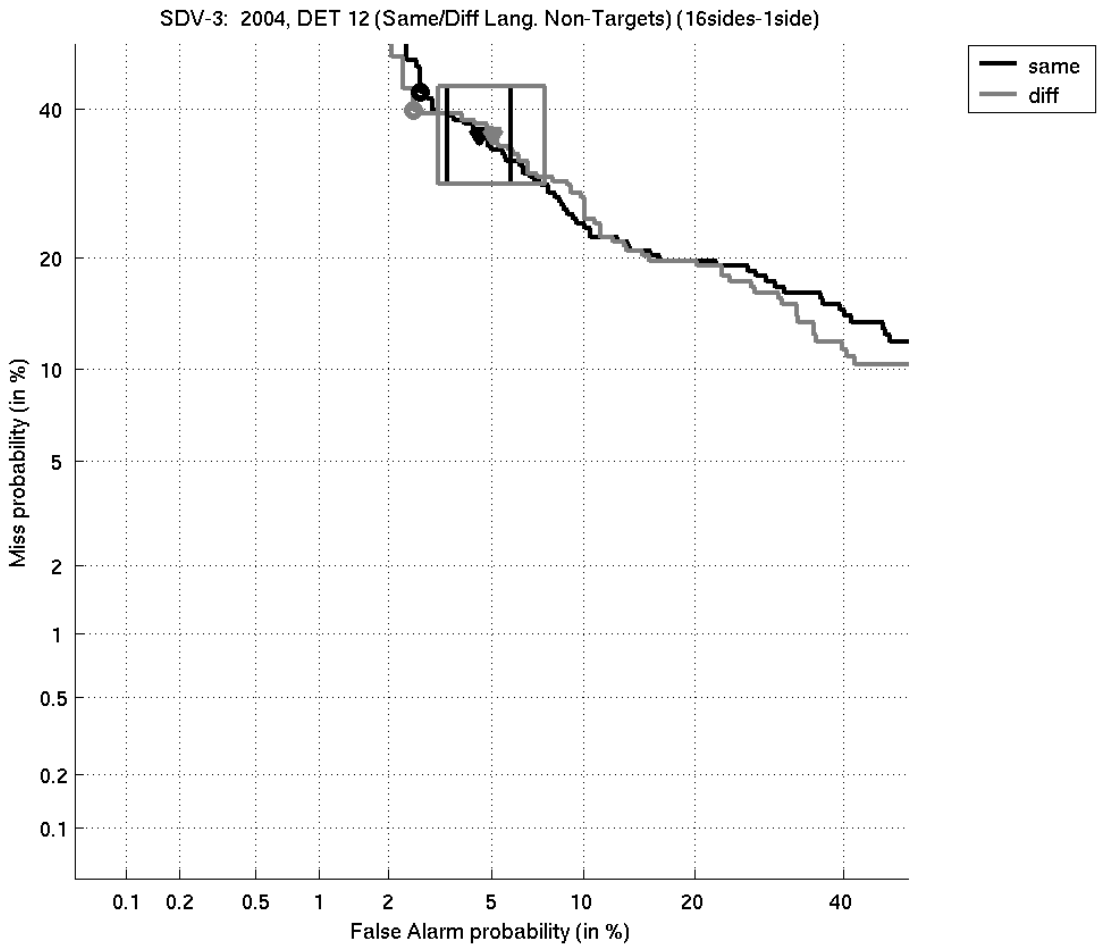


Figure D.20: *DET curves of same and different language non-target trials of SDV_3, using 16 (5 minute) conversation sides for training and 1 conversation side for test segment trials.*

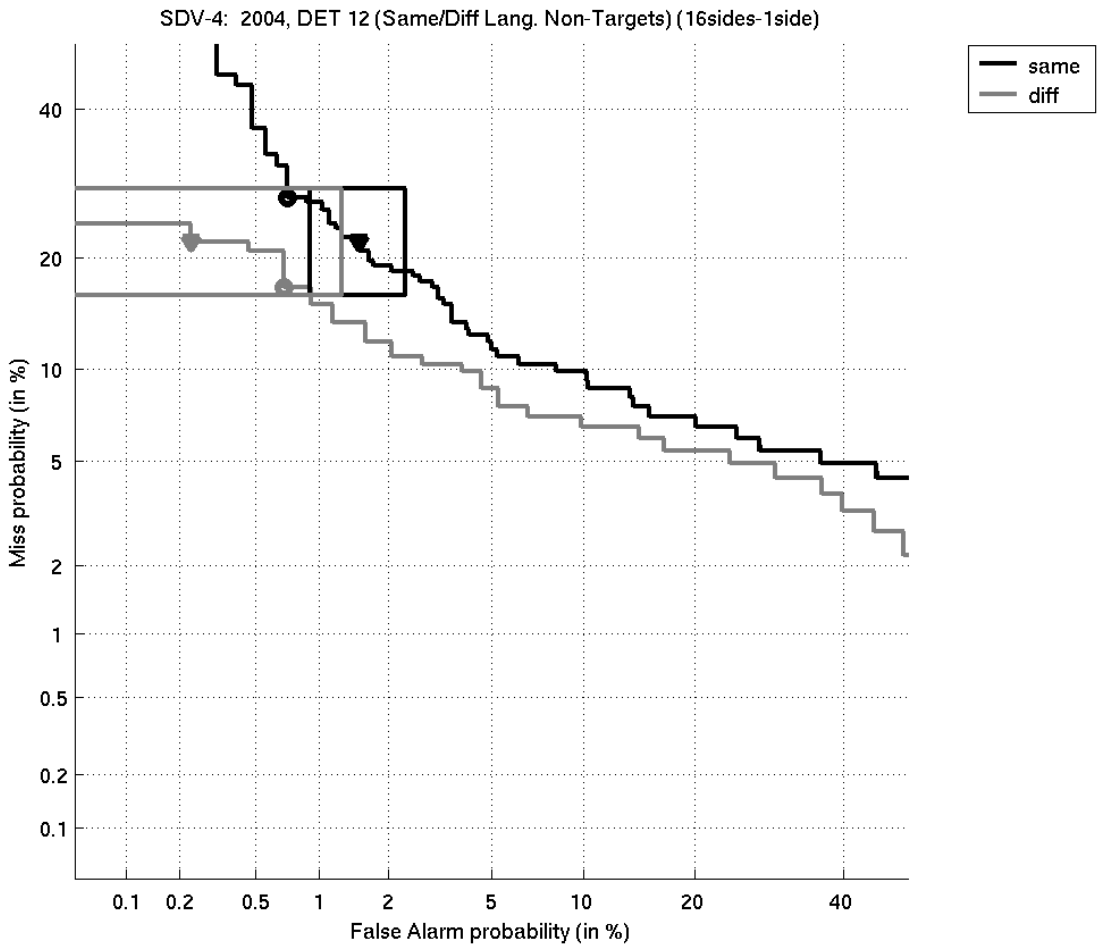


Figure D.21: *DET curves of same and different language non-target trials of SDV_4, using 16 (5 minute) conversation sides for training and 1 conversation side for test segment trials.*

D.2.7 Phone Types

Figure D.22 shows three DET plots where the training data of target trials are collected from cordless phone and the test segment data is collected from 1) regular landline phone 2) cellphone 3) cordless phone. All non-target trials are fixed to include all trials trained on cordless phone data (using 8 conversation sides for training).

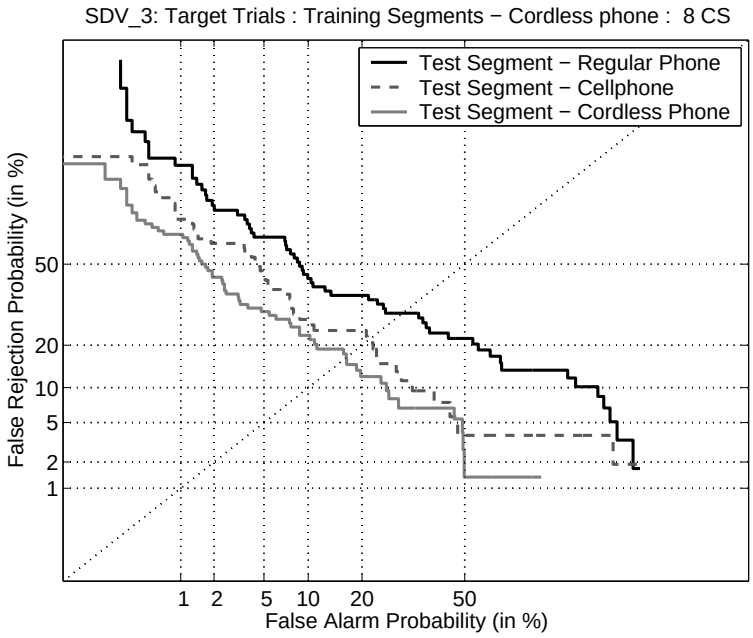


Figure D.22: *DET curves showing the performance of trials where models are trained using 8 conversation sides and* ***cordless phones****.*