Native Accent Classification via I-Vectors and Speaker Compensation Fusion

Andrea DeMarco, Stephen J. Cox

School of Computing Sciences, University of East Anglia, Norwich, England

a.de-marco@uea.ac.uk, s.j.cox@uea.ac.uk

Abstract

We present a comprehensive analysis of the use of I-vector based classifiers for the classification of unlabelled acoustic data as native British accents. We demonstrate the different behaviours of various popular dimensionality reduction techniques that have been previously used in problems such as speaker and language classification. Our results show that a fusion of I-vector based systems gives state-of-the-art performance for unlabelled classification of British accent speech data, reaching ∼81% accuracy.

Index Terms: Accent classification, Ivector representation, Dimensionality reduction, SVM, LDA, Discriminant Analysis, NCA

10.21437/Interspeech.2013-382

1. Introduction

Accents of a language can manifest in different ways: they can have *systemic* differences (different phonemes), *realisational* differences (certain phonemes are always realised differently), *distributional* differences (phoneme distributions are different) or *selectional* differences (particular words are realised differently) [1]. Methods of accent classification that emulate human processes attempt to capture some of these differences using an accurate phonetic transcription of utterances, and this can give high accuracy [2]. However, such a transcription is unlikely to be available in real-world scenarios and phoneme errors in recognition output are likely to make this approach fail. Our approach is to use acoustic data labelled only by accent, without any phonetic labelling, and to use machine-learning techniques to model the above variations implicitly. This may give lower accuracy, but it is a practical approach that enables accent classification to be linked with speaker recognition, which is our ultimate goal.

In this work, we have used the Accents of the British Isles (ABI-1) corpus [3], which is one of the most comprehensive corpora of native accents of a single language that is available. Acoustic-only processing techniques for classification of the ABI-1 corpus have been studied in depth by Hanani et. al. [4]. Traditional methods such as GMM-UBM, GMM-SVM supervector, and GMM-indexed ngram systems, and fusions of these have all been utilised, and the reader is referred to [4] for detailed descriptions of these methods. Fusing all the techniques described in [4] gave an accuracy of 73.6% on 30-second cuts.

The *I-vector* approach has been shown to be an excellent method of modelling speech variability, without any knowledge of the phonetic content of the speech, whilst focusing on variation that is important for discriminating an attribute of the speech that is of interest e.g. speaker, language, channel etc. It currently gives state of the art performance for speaker verification and language identification [5, 6, 7, 8]. In previous work [9], we described an initial experiment with I-vectors for classification of the ABI-1 data. In the work presented here, we show that I-vector based methods can also yield state-of-the-art performance for accent classification with ∼13% absolute classification accuracy improvement over our previous I-vector based classification, and a ∼7% absolute classification accuracy improvement over the previous state-of-the-art acoustic classifier on the same corpus.

2. Use of corpus

The ABI-1 corpus [3] contains data spoken in fourteen different native British accents. Each accent group has ten speakers per gender. In order to ensure speaker-independent accent classification testing, the speakers were divided into three roughly equal sets. Every set contains both male and female speakers, and no speaker from one set is found in any of the other two sets. Two sets contain 98 speakers, and one set contains 84 speakers. For every experiment, the training set is made up of two of these sets, whilst the third set is used for testing. To test all three sets, the groups are transposed three times and results are pooled. In our previous work [9], we used training data from all utterances in the training set, and then tested on the three long passages for each speaker. However, in this work, we use only the three long passages per speaker for both training and testing.

3. System description

3.1. Feature extraction

The feature extraction process is performed as follows: *a*) perform voice activity detection [10]; *b*) extract 13-dimensional MFCC vectors on the speech utterance, with a window of 30ms and a frame rate of 15ms; *c*) convert each MFCC vector into a 49-dimensional shifted delta cepstral (SDC) vector using a 7-1-3-7 SDC parametrization [11]; *d*) warp original MFCC feature vectors to a standard normal distribution with a 3 second time window to minimize effects of channel mismatch [12]; and *e*) concatenate the warped MFCC feature vectors with their respective SDC vectors, to form a final set of 62-dimensional feature vectors.

3.2. Accent total variability

The first uses of total variability and I-vector methods for speech classification were in the area of speaker verification [7, 5]. The I-vector representation was based on the success of the joint factor analysis (JFA) technique. For the purposes of speaker identification, factor analysis is used to construct a low-dimensional subspace, termed the total variability space. This space contains factors of both speaker and channel variability. Unlike JFA, all the variability is contained in a single subspace, whereas each kind of variability is modelled in an explicitly separate subspace in JFA. Once the total variability space is estimated (using training set only), various methods of intersession

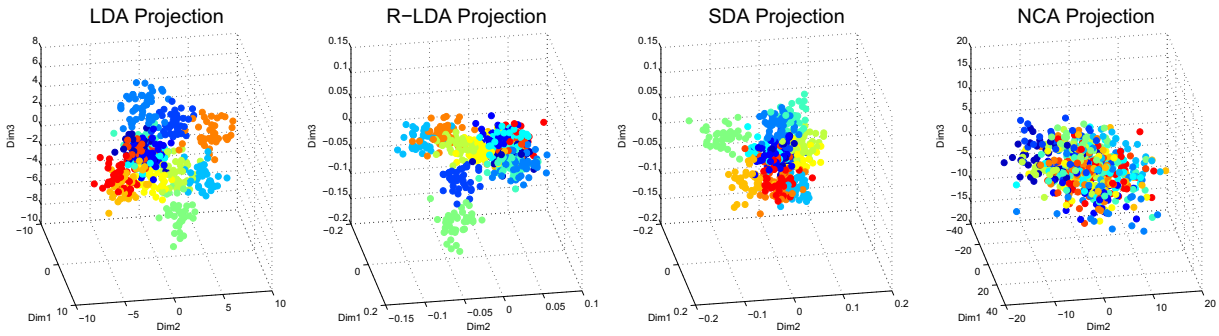


Figure 1: Projections of I-vectors for various dimensionality reduction techniques.

compensation can be performed. For the purpose of speaker identification, one would perform compensation say, on channel effects, to retain only speaker-discriminatory information. The premise of a representation of the data in total variability space is that a universal background model (UBM), trained on data from multiple speakers, can be adapted to a given utterance, creating an utterance-dependent Gaussian mixture model (GMM). The eigenvoice adaptation technique assumes that the matrix T contains speaker and channel variability information. The utterance GMM supervector M is the concatenation of mean vectors of adapted GMM and is obtained as shown in Equation 1.

$$M = m + Tw \tag{1}$$

In this equation, m is the UBM supervector, T is the total variability space matrix, and w is the I-vector, which is a random vector with a normal distribution $\mathcal{N}(0, I)$. The I-vector w is obtained for any given utterance. The method is outlined in full in [5].

The I-vector representation technique has been successfully applied to language recognition [6], and we treat accent recognition as the same problem, although the differences between classes are much finer in accent recognition. The idea is to work on eigenlanguage or eigenaccent adaptation rather than eigenvoice adaptation. In speaker identification, the total variability space T is estimated using all utterances of a particular speaker as belonging to the same class. In language or accent classification, every utterance is considered as coming from a different language or accent class. This is because each utterance has variability due to both speaker differences and language/accent differences.

3.3. Universal background model

Training of the universal background model (UBM) is based on individual training sets (sets of two) as described earlier. We utilized no other corpora in the training of the system. Note that the ABI-1 corpus is small compared to other corpora that have been used for language recognition. About eight hours of data is available for training and testing, and therefore, training is performed on approximately five hours of data, rotated as described in Section 2. If we utilize standard UBM construction techniques of direct estimation, when using a large number of mixture components, there is a high chance of running into problems of components having very small variances (singularities). To circumvent the stability and variance problems, we perform a slower, but more stable UBM construction by first obtaining a VQ codebook via the Linde-Buzo-Gray (LBG) algorithm [13]. The codebook splitting criterion we used was to double the number of centroids at every LBG iteration, and then re-estimate the centroid means via a traditional k-means algorithm until the desired number of centroids is reached. Once the cluster centroids (the VQ codes) are estimated, the covariances and weights of each cluster are estimated. This initial estimation is then passed on to a UBM trainer to perform five iterations of Expectation-Maximization, which outputs the final UBM.

4. Feature compensation methods

After extracting the I-vectors using Equation 1, the accent classes are separated by either using a discriminating projection followed by a suitable classifier, or by using a discriminatory classifier. We analyse four different projection techniques which we shall describe briefly in the subsections below. A sample plot of the I-vectors for accent data (where each accent has a different colour), after performing projection is shown in Figure 1. All the projections we consider are linear, and therefore directly applicable to test data.

4.1. Linear discriminant analysis

Linear discriminant analysis [14] (LDA) is a popular technique to reduce the dimensionality of I-vectors. LDA projects I-vectors into a new subspace of reduced dimensionality that aims to maximize the ratio of between-class variance to the within-class variance, thus optimizing linear separability. Whatever the I-vector dimensionality being utilized, LDA reduces this to one dimension fewer than the number of accent classes.

4.2. Regularized linear discriminant analysis

In the case of high dimensionality feature vectors, LDA suffers from the small sample size problem, and has shortcomings such as the assumption of a common covariance matrix for all classes. There is no reason to consider that all accent classes satisfy this criterion. One way of circumventing this assumption is to assume a separate covariance matrix for each class, leading to quadratic discriminant analysis (QDA). There is, however, an intermediate method between LDA and QDA, proposed by Friedman [15], termed regularized-LDA (R-LDA). In R-LDA, a regularization term is used to shrink the separate class covariance matrices in QDA towards a common covariance as in LDA.

4.3. Semi-supervised discriminant analysis

Semi-supervised discriminant analysis (SDA) was proposed by Cai et. al. [16]. Similarly to RLDA, it also aims to overcome some of the problems with LDA, specifically that of not having enough training samples, and therefore creating an ill-formed projection. The idea of SDA is to use labelled data just like in LDA to maximize class separability, but also to use unlabelled

samples to estimate the intrinsic geometric structure of the data. SDA is designed to estimate a projection that satisfies the LDA objective, but also avoids an 'overfit' in the data projection manifold. This is a very interesting idea for accent classification, since the different speakers in the three test sets can produce very different LDA projections. By using unlabelled test-set points at testing time, we build a smoother manifold, which is more representative of our test data.

4.4. Neighbourhood component analysis

Neighborhood component analysis (NCA) was proposed by Goldberger et. al. [17]. The technique is not part of the family of DA techniques, but is also a popular dimensionality reduction technique, and in various results such as [6], provides better language recognition results when compared to LDA. Unlike typical DA methods, NCA makes no assumptions about the shape of class distributions and the boundaries between them. It tries to utilize the power of k-nearest neighbour (KNN) classification for non-linear boundaries. In contrast with KNN, NCA is designed to learn a distance metric based on the labelled training data, since standard metrics such as Euclidean distance may be ineffective for problems such as language or accent classification. The projection and metric given by NCA minimizes the training error defined using leave-one-out cross validation, and is optimized so that 1-NN classification performs well afterwards.

5. Classification

In this work, we evaluate various dimensionality reduction techniques under three classifier types: cosine kernel support vector machine (SVM), LDA classification and 1-NN classification. Whilst SVMs with a linear kernel and LDA classifiers share something in common, that of finding a linear boundary between classes, linear SVMs are more sophisticated in that the separation rule attempts to find a linear boundary between two parallel hyperplanes in a way that the distance between these hyperplanes is maximized. We can think of linear SVMs as a generalization of LDA classifiers. However, given that we perform dimensionality reduction using in most cases, DA-based projections, it is interesting to see how these methods compare.

5.1. SVM classifier with cosine kernel

A very popular classification technique in I-vector space in the fields of speaker and language identification is the use of support vector machines (SVMs) with a cosine kernel [6, 18]. The cosine kernel is a norm-normalized version of the linear kernel, and for two I-vectors w_1 and w_2 is computed by Equation 2.

$$k(w_1, w_2) = \frac{w_1^t . w_2}{\|w_1\| \|w_2\|} \tag{2}$$

The effect of this cosine kernel over a the standard linear kernel is that the normalization compensates a widening of the data in kernel space, which has a positive effect in the case of i-vectors. In our work we implement a precomputed cosine kernel and build a multi-class SVM classifier system using the LIBSVM toolkit [19]. The SVM can be extended to use speaker-compensated I-vectors after dimensionality reduction using any of the methods we described earlier. If we consider a dimensionality reduction method to have produced a projection matrix A, then the new cosine kernel for two I-vectors w_1 and w_2 is computed by Equation 3.

$$k(w_1, w_2) = \frac{A^t w_1^t . A^t w_2}{\|A^t w_1\| \|A^t w_2\|} \tag{3}$$

5.2. Linear discriminant analysis classifier

We have seen earlier how LDA, and similar DA methods can be used to perform dimensionality reduction by projecting the original input space into a new speaker-compensated subspace consisting of discriminant directions based on the provided class labels, which in our case, are accent classes. LDA itself can be used as a classifier to derive a linear decision surface around the classes in this new subspace. The advantages of this approach are that LDA classification is easier to implement, it is a multi-class classifier, and there are no parameters that need to be tuned via a development data set.

5.3. Length normalization

Length normalization over I-vectors is reported to give performance gains in classification tasks [20]. This process is performed by normalizing every I-vector to a unit vector.

5.4. Fusion

In our experiments, the classifier methods described above are tested on different combinations of I-vector dimensions, dimensionality reduction techniques and UBM component sizes, together with 1-NN classification for NCA dimensionality reduction. Each individual system (up to 84 in total) gives a different classification result. Fusion is performed by majority voting on the outputs of a number of different classifiers. Testing all combinations of classifiers is unfeasible, so a binary genetic algorithm was used to find the best combination of classifiers (although it is not known if this is the optimal combination).

6. Results and discussion

In this section we present our results for the various techniques tested. Each graph has accuracy for four different techniques as a function of UBM size (groups of bars) and I-vector dimensions (individual bars). Figure 2 shows results for a number of UBMs and I-vector sizes, with and without length normalisation. For UBMs of 512 and 1024, length normalization

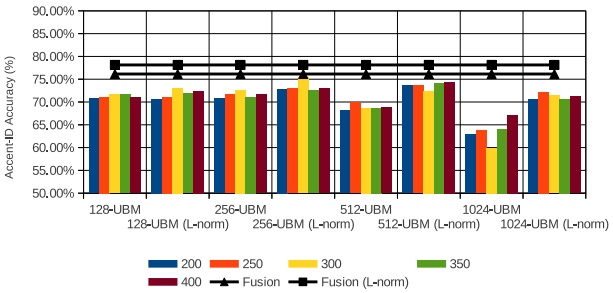
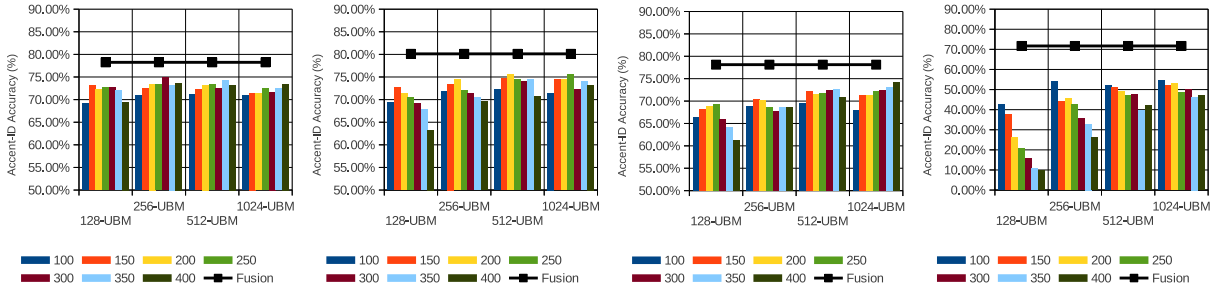


Figure 2: LDA Classification over LDA reduced I-vectors.

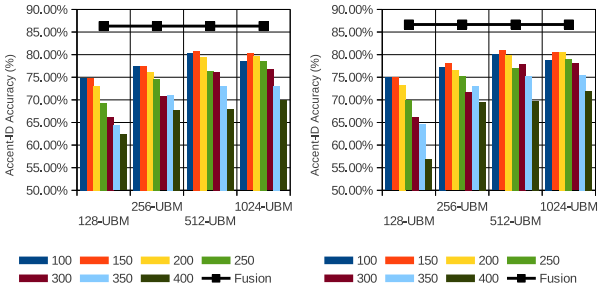
gives statistically significantly better performance than no normalisation (McNemar's test [21] gave p-values in the range [0 0.0037]). It was also found that these results, which are based purely on 30s passages, are better than the results presented in [9], which used utterances of all lengths, including isolated words and short phrases. This suggests that the I-vector approach works best on longer samples, as reported



(a) LDA Classification over R-LDA reduced I-vectors. (b) LDA Classification over SDA reduced I-vectors. (c) SVM Classification over SDA reduced I-vectors. (d) 1-NN Classification over NCA reduced I-vectors.

Figure 3: R-LDA reduction fusion marginally better than LDA reduction fusion. SDA gives the best performance in our experiments. SVM classification has no gain over LDA classification. NCA 1-KNN classifier is the weakest in our tests (note y-axis scale in 3(d)).

in [22, 23, 24]. Fusion for non-normalized and normalized systems reach 76.14% and 78.13% respectively.



(a) LDA Classification (LOSO) over LDA reduced I-vectors. (b) LDA Classification (LOSO) over SDA reduced I-vectors.

Figure 4: With LOSO training SDA reduction outperforms LDA reduction by a very small margin, since the overall gain of SDA is reduced when only data for one speaker is used for online unsupervised training.

Given the gain obtained by length normalization over I-vectors, all further experiments were performed this way. A test based on R-LDA dimensionality reduction using an LDA classifier was performed. Comparing Figure 2 with Figure 3(a), although the individual behaviour of all systems changes slightly, the final fusion results are the same. On the other hand, when we employed SDA for dimensionality reduction (seen in Figure 3(b)), performance improved across the board, giving a fusion result of 80.12%. When using SDA reduction combined with SVM classification, the fusion result was 78.13% (Figure 3(c)), thus confirming LDA classification is better for this classification problem. A fusion of all the classifiers made up of systems from LDA and SDA reduction increases the classification accuracy from 80.12% to 80.70%. NCA fusion on the other hand, performs the weakest (Figure 3(d)). However, when all systems, including NCA based classifiers are fused together, performance is further boosted to 81.05%.

6.1. Leave-one-speaker-out (LOSO) training

In order to test the effect of more training data on classifier performance, we also performed tests where the classifiers were built using all the data available, except for I-vectors from a single speaker. Each speaker was tested individually, and results were pooled. Based on the previous results, we opted for LDA and SDA based classifiers. The results are shown in Figure 4. Again, a fusion of all the classifiers made up of systems from LDA and SDA reduction increases the classification accuracy from 86.67% to 87.13%. This is an encouraging result for the use of accent classification in speaker identification, as individual's accents can be more accurately profiled during enrolment and at testing.

6.2. Comparison of results

A comparison of results in Table 1 shows how I-vector based systems are superior to other systems. Unlike in language identification, NCA dimensionality reduction and a subsequent 1-NN classifier is the poorest performer. SVM classifiers perform well, but are outperformed by the LDA classifier when using SDA, and LDA+SDA fusion based dimensionality reduction.

#	Classifier Type	Acc.
1	GMM-UBM(4096)	56.11%
2	GMM-SVM (4096)	67.72%
3	GMM-uni-gram	60.12%
4	GMM-bi-gram	52.12%
5	Methods #1 to #4 above fused	73.6%
6	NCA Fusion, 1-NN class.	71.70%
7	SDA Fusion, SVM class.	78.13%
8	LDA Fusion, LDA class.	78.13%
9	R-LDA Fusion, LDA class.	78.25%
10	SDA Fusion, LDA class.	80.12%
11	LDA+SDA Fusion, LDA class.	80.70%
12	**LDA+SDA+NCA Fusion, LDA/1-NN class.**	**81.05%**

Table 1: Comparison of classification accuracy of the results obtained in Hanani et. al. [4] (# 1–5) and the results obtained in this work (# 6–12) for 30 second test utterances.

7. Conclusions

In this paper, we have presented a comprehensive investigation of I-vector based classification of accents of the British Isles. The work shows that purely acoustic approaches are capable of giving good accent classification results, despite the fact that accent differences are much finer-grained than language differences. The performance obtained is the highest ever for acoustic classification of the ABI-1 corpus, over $\sim$7% absolute above the previous best. We highlight a number of interesting findings:

- length normalization is advantageous to I-vector systems in this task;
- Neighbourhood component analysis (NCA) followed by 1-NN classification does not perform well in accent classification, unlike in speaker and language classification;
- an I-vector based LDA classifier based on fusion between I-vector systems derived by LDA and SDA dimensionality reduction is better than SVM classification.

Future work will investigate the use of accent profiling in speaker identification.

8. References

[1] J.C. Wells, *Accents of English: An Introduction*, Accents of English. Cambridge University Press, 1982.

[2] Mark Huckvale, "ACCDIST: An Accent Similarity Metric for Accent Recognition and Diagnosis," in *Speaker Classification (2)*, 2007, pp. 258–275.

[3] S.M. D'Arcy, J.M. Russell, S.R. Browning, and M.J. Tomlinson, "The Accents of the British Isles (ABI) Corpus," in *Modelisations pour l'Identification des Langues. MIDL Paris*, 2005, pp. 115–119.

[4] Abualsoud Hanani, Martin J. Russell, and Michael J. Carey, "Human and computer recognition of regional accents and ethnic groups from British English speech," *Computer Speech and Language*, 2012.

[5] N. Dehak, P.J. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 19, no. 4, pp. 788 –798, May 2011.

[6] Najim Dehak, Pedro A. Torres-Carrasquillo, Douglas A. Reynolds, and Réda Dehak, "Language recognition via i-vectors and dimensionality reduction," in *INTERSPEECH*, 2011, pp. 857–860.

[7] Najim Dehak, Réda Dehak, Patrick Kenny, Niko Brümmer, Pierre Ouellet, and Pierre Dumouchel, "Support vector machines versus fast scoring in the low-dimensional total variability space for speaker verification," in *INTERSPEECH*, 2009, pp. 1559–1562.

[8] David Martínez, Oldrich Plchot, Lukás Burget, Ondrej Glembek, and Pavel Matejka, "Language recognition in ivectors space," in *INTERSPEECH*, 2011, pp. 861–864.

[9] Andrea DeMarco and Stephen J. Cox, "Iterative classification of regional british accents in i-vector space," in *Proceedings of Symposium on Machine Learning in Speech and Language Processing (MLSLP 2012)*, September 2012.

[10] Jongseo Sohn, Student Member, Nam Soo Kim, and Wonyong Sung, "A statistical model-based voice activity detection," *IEEE Signal Process. Lett*, vol. 6, pp. 1–3, 1999.

[11] Pedro A. Torres-Carrasquillo, Elliot Singer, Mary A. Kohler, and J. R. Deller, "Approaches to language identification using gaussian mixture models and shifted delta cepstral features," in *Proc. ICSLP 2002*, 2002, pp. 89–92.

[12] Jason Pelecanos and Sridha Sridharan, "Feature Warping for Robust Speaker Verification," in *IEEE Odyssey*, 2001, pp. 213–218.

[13] Y. Linde, A. Buzo, and R. Gray, "An algorithm for vector quantizer design," *Communications, IEEE Transactions on*, vol. 28, no. 1, pp. 84 – 95, Jan 1980.

[14] R. A. Fisher, "The use of multiple measurements in taxonomic problems," *Annals of Eugenics*, vol. 7, no. 2, pp. 179–188, 1936.

[15] Jerome H. Friedman, "Regularized Discriminant Analysis," *Journal of the American Statistical Association*, vol. 84, no. 405, pp. 165–175, 1989.

[16] Deng Cai, Xiaofei He, and Jiawei Han, "Semi-supervised discriminant analysis," in *Proc. Int. Conf. Computer Vision (ICCV'07)*, 2007.

[17] Jacob Goldberger, Sam Roweis, Geoff Hinton, and Ruslan Salakhutdinov, "Neighbourhood components analysis," in *Advances in Neural Information Processing Systems 17*. 2004, pp. 513–520, MIT Press.

[18] Mohammed Senoussaoui, Patrick Kenny, Najim Dehak, and Pierre Dumouchel, "An i-vector extractor suitable for speaker recognition with both microphone and telephone speech," in *Odyssey*, 2010.

[19] Chih-Chung Chang and Chih-Jen Lin, "LIBSVM: A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, pp. 27:1–27:27, 2011.

[20] Daniel Garcia-Romero and Carol Y. Espy-Wilson, "Analysis of i-vector length normalization in speaker recognition systems," in *INTERSPEECH*, 2011, pp. 249–252.

[21] Larry Gillick and SJ Cox, "Some statistical issues in the comparison of speech recognition algorithms," in *Acoustics, Speech, and Signal Processing, 1989. ICASSP-89., 1989 International Conference on*. IEEE, 1989, pp. 532–535.

[22] Ahilan Kanagasundaram, Robbie Vogt, David B. Dean, Sridha Sridharan, and Michael W. Mason, "i-vector based speaker recognition on short utterances," in *Interspeech 2011*, Firenze Fiera, Florence, August 2011, pp. 2341–2344, International Speech Communication Association (ISCA).

[23] Anthony Larcher, Pierre-Michel Bousquet, Kong-Aik Lee, Driss Matrouf, Haizhou Li, and Jean-François Bonastre, "I-vectors in the context of phonetically-constrained short utterances for speaker verification," in *ICASSP*, 2012, pp. 4773–4776.

[24] Wei Rao and Man-Wai Mak, "Alleviating the small sample-size problem in i-vector based speaker verification," in *ISCSLP*, 2012, pp. 335–339.