Evaluation of the Macquarie Meeting Room Speaker Diarisation System

Steve Cassidy

Centre for Language Technology
Macquarie University
Sydney
Steve.Cassidy@mq.edu.au

Abstract

This paper describes the methodology for evaluation of a meeting room speaker diarisation system submitted to the 2004 NIST Rich Transcription evaluation. The architecture of the Macquarie submission is described and the results of a detailed post-hoc evaluation of the system are presented.

1. Introduction

In spring 2004 I took part in the NIST meeting room speech rich transcription evaluation, RT04s, submitting results for the speaker diarisation task. Since limited time was available to build the full diarisation system, no thorough evaluation of the components was possible and only the overall results were published in the earlier proceedings. This paper describes the system and develops a framework for the more detailed analysis and evaluation of the different components.

The speaker diarisation task involves segmenting a meeting recording into speaker turns, determining the number of speakers present and providing a speaker label for each turn. No prior knowledge of the number of speakers or of their identity is provided. In the data used for the NIST evaluation, all recordings were made from distant microphones placed on the table during the meeting. In some cases multiple microphones were available although their geometry was not known.

The components of the Macquarie diarisation system are as follows:

- Speech/silence classifier – identifies speech segments in the meeting.
- Speaker segmentation – segments speech segments into speaker turns.
- Speaker clustering – identifies the number of speakers and builds speaker models for each.
- Speaker identification – labels each speaker turn with a speaker label using the speaker models.

The design of this system was very ad-hoc since it was put together at short notice from ’spare parts’ in order to be able to take part in the NIST RT04s evaluations. We had previously done some work on segmenting meeting recordings (Cassidy and Watson 2002; Watson and Cassidy 2003) but this was limited to finding segment boundaries rather than identifying the speakers involved. The goal of the submission was to develop a baseline system which would reveal some of the problems involved in the diarisation task and serve as a base on which to further develop the individual components. Given the time available it was only feasable to develop fairly simple components and a baseline of simple gaussian based models was chosen as being easy to implement and, hopefully, capable of a useful level of performance.

While the project was successful and we were able to submit results to NIST for the RT04s evaluation, the overall performance of the system was not particularly good. While developing the system it was hard to see which component to blame for poor performance since each stage served as input to the next. The obvious response is to evaluate each stage in turn; while this was not possible during the initial development, we have subsequently developed a series of evaluations to assist further development. This paper describes this evaluation program and reports on the performance of the system in more detail than the paper submitted to the RT04s workshop (Cassidy 2004). The paper also considers alternative implementations of some of the stages.

2. Evaluation

The overall diarisation system is evaluated according to an error rate based on the fraction of speaker time that is not attributed correctly to a speaker (Fiscus and Garofolo 2004). While this gives a good overall view of the performance of the system, it is useful to develop a number of more fine-grained measures of performance to evaluate the different components of the system described above. Since the design of the system is inherently a cascade of processes it is useful to evaluate each stage without interference from the performance of earlier stages. To achieve this we rely on already labelled data with which we can compare the results of each stage and from which we can generate known correct input to the next stage.

The data used for this evaluation is the devtest data set distributed prior to the RT04s evaluation by NIST. This consists of two recordings each from ICSI, LDC, CMU and NIST meeting room data collections. A ten minute segment of each meeting is targetted for testing purposes and a full speaker turn annotation is available which includes regions of overlap between speakers. In some cases, recordings are available from more than one distant (desktop) microphone but the Macquarie system only used the nominated *most central* microphone (the NIST **sdm** condition). The evaluation results presented in this paper are taken from four of

Proceedings of the 10th Australian International Conference on Speech Science & Technology
Macquarie University, Sydney, December 8 to 10, 2004. Copyright, Australian Speech Science & Technology Association Inc.

Accepted after abstract only review

these recordings, one from each recording site.

2.1. Speech/Silence

Speech silence segmentation can be seen as a special case of the dialogue segmentation problem but in our system we found it advantageous to remove silence segments in a preliminary stage. This stage is easily evaluated by counting the number of frames of data misclassified relative to the total number of frames of speech or silence.

$$error_{sil} = 100 * \frac{n_{sil \rightarrow speech}}{n_{sil}} \tag{1}$$

$$error_{speech} = 100 * \frac{n_{speech \rightarrow sil}}{n_{speech}} \tag{2}$$

Where $n_{sil \rightarrow speech}$ is the number of frames of silence classified as speech and n_{sil} is the total frames of silence. In this application, it is very important not to miss speech frames and so our system might seek to minimise $error_{speech}$ even if it is at the expense of $error_{sil}$.

Our RT04s system used a simple RMS/Zero crossing based speech detector with a threshold that was adjusted based on the mean of RMS and ZCR of each recording. Frame by frame speech/silence decisions are aggregated so that very short regions of either do not trigger a change of state. Two parameters can be varied in the implentation: the threshold value and the relative weight of the ZCR.

The results were compared with a hand-labelled annotation of the significant silence regions in the recording. This labelling sought to label all silence regions over about 0.3 seconds in length

The results of a number of runs of the algorithm are shown in Table 1. It can be seen that a threshold of 5 and ZCR weight of 1.0 gives the minimum missed speech frames ($error_{sil}$) while retaining relatively low false alarm rate ($error_{speech}$). This improves on the values used in the RT04s submission (threshold 7, ZCR weight 2) which was arrived at by visual examination of a small number of segmentation results. However, as might be expected with this naive algorithm, both error measures are very high – one in three frames of speech will be overlooked. We are currently investigating model based techniques such as support vector machine classifiers to help make these decisions.

2.2. Speaker Segmentation

The speaker segmentation stage accepts segments of speech and finds potential speaker change points within them. The advantage of a prior speech/silence detection stage is that the segments are generally much shorter and many will contain only one speaker turn. Speaker segmentation can be evaluated as described in our earlier paper (Cassidy and Watson 2002) where we measured two kinds of error: **False Positive** errors are automatically detected boundaries that occur within a speaker turn, **Missed Boundary** errors are speaker turn boundaries that aren't detected by our algorithm.

$$error_{fp} = 100 * \frac{n_{false}}{n_{auto}} \tag{3}$$

$$error_{missed} = 100 * \frac{n_{missed}}{n_{boundaries}} \tag{4}$$

Silence Threshold	ZCR Weight	$error_{sil}$	$error_{speech}$
5	0.5	43	15
5	1.0	31	12
5	2.0	54	9
5	4.0	73	4
6	0.5	63	25
6	1.0	52	15
6	2.0	36	12
6	4.0	54	9
7	0.5	60	33
7	1.0	76	16
7	2.0	54	15
7	4.0	53	9

Table 1: Results from a number of runs of the speech/silence algorithm under different condition. Error scores are averaged over four ten minute recordings.

Where n_{false} is the number of false positive boundaries, n_{missed} is the number of boundaries not found by the system, $n_{boundaries}$ is the total number of true boundaries and n_{auto} is the total number of automatic boundaries. A boundary is deemed to be correct if it is within 500ms of a real boundary; similarly a real boundary is deemed to have been missed if there is no automatic boundary placed within 500ms.

The first kind of errors are not too serious in this context since they just mean that a speaker turn has been too finely segmented. Missed boundary errors are more important since they represent missed acoustic changes.

The implemented segmentation algorithm is based on the Bayesian Information Criterion or BIC (Chen and Gopalakrishnan 1998; Zhou and Hansen 2000). This algorithm evaluates potential break points based on the goodness of fit of either a single model or two seperate models. In the simple case, as implemented here, the models compared are simple multivariate gaussian models although in other work gaussian mixture models have been used. The algorithm has a single parameter, λ, which is effectively a threshold value for the ratio of the probability of the data given one vs. two models after correction for the complexity of the models. Ideally λ is 1.0 since the BIC method is intended to be threshold free but in practice this parameter can be tuned to give best performance for a given task.

Table 2 show the results of evaluating this part of the system at different settings of the λ parameter. Scores are averaged over all four evaluation recordings. It can be seen that there is a large proportion of false positives in all cases. Lowering the lambda threshold generates more hypotheses and so leads to a higher false positive rate but lowers the missed rate. An optimal setting seems to be $\lambda = 0.9$ which differs from the $\lambda = 1.1$ used in the submitted RT04s system. In general though, we can that this very large number of false positive errors will lead to shorter speaker segments which might be harder to recognise.

λ	$error_{missed}$	$error_{fp}$	n_{auto}
0.7	12	52	2294
0.8	12	52	2254
0.9	12	51	2196
1.0	15	50	2089
1.1	18	48	1933
1.2	22	45	1724

Table 2: Error scores for the speaker segmentation part of the system showing the effect of the λ threshold parameter on overall performance.

2.3. Speaker Clustering

The speaker clustering stage is perhaps the most difficult part of the speaker diarisation problem. Given the nature of the meetings being processed, there is often a large amount of speech from one speaker with much smaller contributions from others. With no prior indication of the number of speakers we must evaluate different numbers of clusters, choosing the one which seems to best account for the data.

The evaluation of this stage can be performed on the overall performance of the clustering algorithm and the number-of-clusters determination and in fact this is the goal of the main NIST evaluation metric. Alternately and more usefully in development of the system we can measure the performance of the clustering algorithm separately – assuming that we know hoe many clusters are there to be found. One metric for evaluating clustering methods is suggested by (Dhillon, Fan, and Guan 2001) and consists of two measures: cluster *purity* and *entropy*. These are defined as:

$$purity_i = \frac{1}{n_i} \max n_i^h \tag{5}$$

$$entropy_i = \frac{1}{\log c} \sum_{h=1}^{c} \frac{n_i^h}{n_i} \log \frac{n_i^h}{n_i} \tag{6}$$

where n_i is the number of tokens in cluster i, n_i^h is the number of tokens in cluster i which belong to the reference speaker h and c is the number of reference speakers. To summarise performance over a number of clusters we report the average values of entropy and purity. Average entropy is just the arithmetic mean of the entropies of the individual clusters. Average purity is calculated as a weighted mean with each purity value multiplied by the number of tokens in the cluster:

$$purity = \frac{1}{N} \sum_{i=1}^{C} purity_i * n_i \tag{7}$$

Where N is the total number of tokens being clustered and C is the number of clusters (which in our evaluation is the same as c the number of reference speakers).

Cluster purity measures the ratio of the size of a cluster to the size of its dominant class; a value of 1 means that this cluster contains just one reference speaker. Entropy measures the distribution of reference speakers among clusters and will be close to zero for clusters containing just one reference speaker but close to 1 for clusters which are a uniform mixture of different speakers.

Our RT04s system used a hierarchical clustering method using the means and variances of the input parameters as features; effectively, each token is modelled as a single gaussian distribution and the inter-token distances are calculated by comparing the means and variances of these models. The distance measure used was a mahalanobis distance between the token models. Only segments longer than 1.5 seconds were included in the clustering process; the reasoning being that segments shorter than this would be difficult to characterise with a gaussian model.

The results of evaluating our clustering algorithm as described above is shown in Table 3. It is clear from these results that this clustering method is not doing a particularly good job of grouping like tokens together. The average purity of the clusters generated is around 0.5 meaning that only around half of the cluster contents come from the same speaker. The entropy lies over 0.5 indicating that the clusters generally contain a few speaker's data.

Meeting	Speakers	Purity	Entropy
ICSI-20010322-1450	7	0.48	0.63
CMU-20020320-1500	4	0.67	0.57
LDC-20011116-1400	3	0.59	0.74
NIST-20020214-1148	6	0.71	0.46

Table 3: Cluster purity and entropy for the Gaussian feature set.

The results in Table 3 indicate the performance of the submitted RT04s system which did not have any tuneable parameters for this part of the algorithm. While these results seem poor it's hard to evaluate them in isolation. Subsequent experiments have been done with an alternate characterisation of each token using the so-called Fisher Kernel (Smith and Gales 2002). This seeks to model the range of variability in the parameters used to characterise the signal by finding the gradient of the data with respect to the parameters of a model of the data. In our implementation, a simple gaussian model is used again and the gradient of the data with respect to the means and variances (diagonal of the covariance matrix) are used as features. The model of the data was a simple gaussian model trained on around 10% of the speaker turns from each recording selected at random. The results of applying the hierarchical clustering algorithm using the same distance metric on the Fisher Kernel data are shown in Table **??**. They show a slight improvement in purity in all except the ICSI meeting and a larger change in entropy especially in the case of the LDC meeting.

While the purity results for the Fisher Kernel are similar to those with the Gaussian model the entropy results are lower indicating that each cluster tends to contain fewer speakers when using these features. We hope that the extension of this method to more complex models of the data will lead to significantly better results.

Meeting	Speakers	Purity	Entropy
ICSI-20010322-1450	fisher	0.44	0.41
CMU-20020320-1500	fisher	0.67	0.47
LDC-20011116-1400	fisher	0.59	0.69
NIST-20020214-1148	fisher	0.69	0.22

Table 4: Cluster purity and entropy for the Fisher Kernel feature set.

2.4. Speaker Identification

The final stage of the process is to use speaker models trained on the already clustered speech to identify all of the speaker turns in the meeting recording. In our system this was necessary because the previous speaker clustering stage was carried out only on utterances longer than two seconds. This problem is essentially the same as the well understood speaker identification problem that has been addressed many times and whose evaluation is well understood. The main difference is that the speakers being identified are not known outside of the recording being labelled, hence any speaker models must be trained (or adapted) on only the data in a given meeting.

Our system used a very simple Gaussian model for each speaker trained on the clustered speech as described above. This kind of model is known to be unlikely to work very well especially in this context of free text, distant microphone recordings. Since the speakers are not known beforehand we do not have existing speaker models to match unknown segments against. The previous stage in the system clustered segments longer than 1.5 seconds leaving shorter segments unlabelled. In this stage, the system builds speaker models from this already labelled speech and uses this to label the shorter segments. On average, more than half of the segments were below this length limit.

Evaluation of speaker identification systems is simply measured by the proportion of classification errors generated. Table 5 shows the results from our system. The overall error rate of NN% is very high.

Meeting	% Error	Number Turns
ICSI-20010322-1450	42	234
CMU-20020320-1500	34	186
LDC-20011116-1400	29	208
NIST-20020214-1148	45	217
Mean	37	

Table 5: Speaker ID error scores

3. Overall Performance

Evaluation of the whole system is based on the evaluation metric defined by NIST for this task. They calculate an overall speaker diarisation error score based on the fraction of speaker time that is not attributed correctly to a speaker (Fiscus and Garofolo 2004). The scoring metric uses an alignment algorithm to maximise the correspondence between reference and automatic speaker labels. Also reported are missed speaker time, the proportion of speech which was not attributed to any speaker and false alarm speaker time, the proportion of silence which was attributed to some speaker.

The results in Table /reftab:overall compare the evaluation of the submitted system on the devtest data with that achieved using the settings derived above for the various parameters of the system. Clearly no significant change has occurred except for an increase in missed speaker time. The main contributor to the overall score is the speaker error time which can be attributed to the speaker clustering and identification stage. The results below are given for a fixed number of clusters (3), the speaker clustering algorithm tends to choose a larger number of clusters which lead to a much higher overall error score. In fact, the heuristic of always choosing three speakers gives a much better overall score than any cluster evaluation metric that has been tried.

	RT04s	Current
Missed Speaker Time	44.9	48.1
F Alarm Speaker Time	1.5	2.3
Speaker Error Time	22.9	19.1
Overall Error	69.3	69.5

Table 6: Summary of overall evaluation results submitted to the RT04s evaluation in comparison with the current system. All figures are percentages.

4. Conclusion

The goal of this paper is a post-hoc re-evaluation of the components of the Macquarie submission to the NIST RT04s evaluation. The paper has described a number of evaluation methods that can be used independantly on the components of the system to give independant measures of performance. These can then be used to maximise the performance of the overall system.

It is clear from this evaluation that the current system performs only at a very basic level; however, the point of the exercise is to highlight the major areas of weakness and the evaluation succeeds in doing this. We are now working on putting improved algorithms into place, hoping to be ready for the next round of NIST rich transcription evaluations in 2005.

References

Cassidy, S. (2004). The macquarie speaker diarisation system for rt04s. In J. Fiscus and C. Laprun (Eds.), *Proceedings of the NIST Rich Transcription Workshop*.

Cassidy, S. and C. Watson (2002). Detecting backchannel intrusions in multi-party teleconferences. In *Proceedings of the 9th Australian International Speech Science and Technology Conference*, Melbourne. Australian Speech Science and Technology Association.

Chen, S. and P. Gopalakrishnan (1998, Feb). Speaker, environment and channel change detection and clustering via the bayesian information criterion. In *Proceedings of Broadcast News Transcription and Understanding Workshop*.

Dhillon, I., J. Fan, and Y. Guan (2001). *Data Mining for Scientific and Engineering Applications*, Chapter Efficient Clustering of Very Large Document Collections. Kluwer. http://www.cs.utexas.edu/users/yguan/papers/effclus.

Fiscus, J. G. and J. S. Garofolo (2004). Spring 2004 (rt-04s) rich transcriptionmeeting recognition evaluation plan. Technical report, NIST. `http://www.nist.gov/speech/tests/rt/rt2004/spring/documents/rt04s-meeti%ng-eval-plan-v1.pdf`.

Smith, N. and M. Gales (2002). Speech recognition using svms. In T. Dietterich, S. Becker, and Z. Ghahramani (Eds.), *Advances in Neural Information Processing Systems*, Volume 14. MIT Press. `http://mi.eng.cam.ac.uk/˜mjfg/publications.html`.

Watson, C. and S. Cassidy (2003, April). Speaker change detection in multi-party meetings. In *Proceedings of the Eighth Western Pacific Acoustics Conference*, Melbourne.

Zhou, B. and J. Hansen (2000). Unsupervised audio stream segmentation and clustering via the bayesian information criterion. In *Proceedings of ICSLP*, Beijing, China.