THE RT04 EVALUATION STRUCTURAL METADATA SYSTEMS AT CUED

M. Tomalin and P.C. Woodland

Cambridge University Engineering Department, Trumpington Street, Cambridge, CB2 1PZ, UK.
Email: {mt126,pcw}@eng.cam.ac.uk

ABSTRACT

This paper describes the development of the Cambridge University Fall 2004 Rich Transcription evaluation structural metadata systems. Details are given concerning the systems that were constructed for the Conversational Telephone Speech Slash Unit Boundary Detection, Filler Word Detection, and Interruption Point detection tasks, as well as for the Broadcast News Slash Unit Boundary Detection task. The systems described all used adapted versions of the same generic architecture. The use of large training data sets for the Slash Unit Boundary Detection task is considered, and development and evaluation results are given for all the systems.

1. INTRODUCTION

As defined within the EARS program, Structural MetaData Extraction (SMD) requires (i) the segmentation of an input speech signal into sentence-like units, and (ii) the identification of specific regions in the input signal which, if necessary, can be extracted from the resulting speech transcription without a significant loss of informational content [1, 2]. Within the EARS Fall 2004 Rich Transcription (RT-04) evaluation framework, four structural metadata extraction tasks were defined for two different domains. The four tasks were Edit Word Detection (EWD), Filler Word Detection (FWD), Interruption Point Detection (IPD), and Slash Unit Boundary Detection (SUBD), while the two domains were Conversational Telephone Speech (CTS) and Broadcast News (BN).

Essentially, the various SMD tasks require specific kinds of metadata to be identified in a given speech signal using input audio files and their corresponding Speech-To-Text (STT) system output files. More specifically, the FWD task requires semantically bleached discourse-structuring elements such as *um* and *anyway* to be identified; the IPD task requires interruption points to be located in the context of edit disfluencies such as repetitions and restarts, while the SUBD task requires the boundaries that separate sentence-like units to be identified. Once these portions of the audio have been identified, it is possible to produce speech transcriptions that are easier to read since the identified structural metadata events enable the transcriptions to be divided into quasi-sentences; non-information bearing elements such as fillers can be removed; and the points at which the speech becomes disfluent can be indicated. Recent research has suggested that 'cleaned-up' transcriptions of this kind can significantly enhance the readability of automatically generated speech system output [3]. In addition, apart from facilitating readability, it is also possible that speech transcriptions which contain metadata information could benefit downstream processing tasks such as automatic machine translation [4].

For the previous RT-03f evaluation, the Cambridge University Engineering Department (CUED) submitted SMD system output for a CTS SUBD system [5] [6]. By contrast, for the RT-04 evaluation, CUED submitted SMD system output for the CTS FWD, CTS IPD, CTS SUBD, and BN SUBD tasks, and, of these, the FWD, IPD, and BN SUBD systems were constructed specifically for the RT-04 evaluation. The CUED SMD systems all share the same basic architecture in which task-specific Language Models (LMs) are combined with Prosodic Feature Models (PFMs) in a lattice-based decoding framework. This paper describes the development of the CUED SMD systems that were constructed for the RT-04 evaluation [7], and the basic format of the paper is as follows. The SMD tasks are defined in more detail in Section 2, and a general overview of the CUED SMD system architecture is presented in Section 3. The various sets of training and development data that were used in the experiments reported in this paper are described in Section 4, and, in Section 5, the general PFM architecture is discussed. In Section 6, the CUED SMD systems are presented in detail, while, in Section 7, various system development results are given. The evaluation results for the various systems described are summarised in Section 8, and the main conclusions outlined in Section 9.

2. THE SMD TASKS

2.1. Slash Unit Boundary Detection

Slash Units (SUs) are sentence-like units. Including information about SU boundaries in a speech transcription can improve readability by facilitating automatic punctuation generation. The SUBD task requires each SU endpoint to be detected in the input signal. An SUBD system must output a start time and duration for each SU, and, in addition, the SUs must be subclassified into one of the following subtypes: *statement, incomplete, question*, and *backchannel*.[1] The primary scoring metric for the SUBD task sums the number of SU boundary insertion, deletion, and substitution errors in the system output file, when compared to the reference file, and divides this sum by the number of SUs boundaries in the reference file.[2] This produces the primary error rate. Full information about the SUBD task and the scoring metric used for RT-04 can be found in [7].

This work was supported by DARPA grant MDA972-02-1-0013. The paper does not necessarily reflect the position or the policy of the US Government and no official endorsement should be inferred.

[1] A more detailed definition of SUs is given in [8] (Section 4).

[2] In RT-03f, substitution errors were not included in the scoring metric numerator.

2.2. Filler Word Detection

The FWD task requires the regions of the input signal that contain filler words to be detected and subclassified into one of the following subtypes: *filled pause* (e.g., *uh, um*), *discourse marker* (e.g., *anyway*, *I mean*), or *explicit editing term* (i.e., a filler word that occurs in the context of an edit disfluency).[3] The FWD system must specify the start time and duration of all regions of the input signal that contain filler words. The primary scoring metric for the FWD task sums the number of filler insertion, deletion, and substitution errors in the system output file, when compared to the reference file, and divides this sum by the number of filler words in the reference file. More detailed information about the FWD task and the scoring metric used for RT-04 can be found in [7].

2.3. Interruption Point Detection

IPs occur when a speaker, who has been speaking fluently, becomes disfluent. Consequently, IPs are located in edit disfluencies. The IPD task requires the time in the input signal when the IP occurs to be specified,[4] and, for the RT-04 evaluation, the IPs did not have to be subclassified. The primary scoring metric for the IPD task sums the number of insertion and deletion errors in the system output file, when compared to the reference file, and divides this sum by the number of IP events in the reference file. The IPD task and the scoring metric used for RT-04 is defined in detail in [7].

3. GENERAL SMD SYSTEM ARCHITECTURE

All of the CUED SMD systems developed for the RT-04 evaluation, for both the CTS and BN domains, used the same general architecture. The main components of this framework are:

- task-specific language models (LMs)
- task-specific Prosodic Feature Models (PFMs)
- a lattice-based 1-Best Viterbi decoding framework

The generic architecture for the CUED SMD systems is given in Fig. 1.

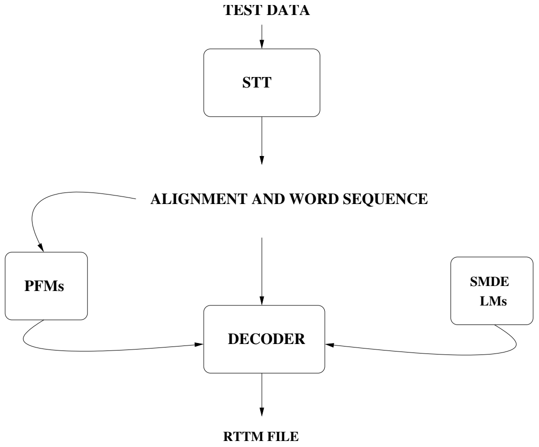

Figure 1: Generic CUED SMD System Architecture

As Fig.1 indicates, the SMD systems use the STT output, which provides hypothesised token sequences and timing information for the input audio files. The SMD LMs and PFMs are free-standing models constructed using the available training data (see Section 4). A lattice is created for each input audio file and the likelihoods obtained from the PFMs are associated with the arcs of the lattices which are then expanded using the LMs and HTK Tools [9]. The 1-Best Viterbi path through each lattice is output, and these paths consist of STT token sequences into which SMD event information has been inserted automatically [10] [11] [5] [6]. These files are then converted into scorable rttm files. As detailed in Section 7, this basic architecture was modified for each of the SMD tasks described in Section 2.

4. DATA USED IN EXPERIMENTS

4.1. CTS Training Data

Four sets of training data were used while building the CUED CTS SMD systems, and these data sets are specified in Table 1. The ctsrt04, ctsrt04_v1.0, and ctsrt03 data sets were prepared by the Linguistic Data Consortium (LDC) for the EARS project. The ctsrt04_v1.0 consists of the same training data files as the ctsrt04 data set; the difference is that the ctsrt04_v1.0 data was annotated using an earlier version of the EARS MDE annotation specification [8]. The WordWave (WW) data had been prepared as training data for the CTS STT RT-04 evaluation. However, since it contained punctuation marks, it was possible to map this data so that it approximated the V6.2 EARS MDE annotation specification for the SUBD task [8]. The basic mapping for the WW data was as follows: all fullstops and commas were mapped to statement SU boundaries, while all question marks were mapped to question SU boundaries.

Name	**ctsrt04**	**ctsrt04_v1.0**	**ctsrt03**	**WW**
Epoch	2004	2004	2003	2004
Released	07/09/04	04/06/04	2003	2004
Spec	V6.2 (v1.1)	V6.2 (v1.0)	V5	mapped
Hours	c.40	c.40	c.30	c.1800
Tasks	all SMD	all SMD	all SMD	SUBD

Table 1. Summary of CTS training data sets used for SMD system development

4.2. CTS Development Data

Three development data sets were used during the process of CTS SMD system development, and these are specified in Table2:

Name	**ctsdev03**	**ctseval03**	**ctsdev04**
Epoch	2003	2003	2004
Spec	V6.2 (v1.1)	V6.2 (v1.1)	V6.2 (v1.1)
Hours	c.1.5	c.1.5	c.3

Table 2. Summary of CTS dev data sets used for SMD system development

The ctsdev03 and ctseval03 data sets constituted the development and evaluation data sets for the RT-03f evaluation [2], while the ctsdev04 data set was prepared as the development set for the RT-04 evaluation.

[3] A more detailed definition of filler words is given in [8] (Section 2).

[4] A more detailed definition of IPs is given in [8] (Section 3).

4.3. BN Training Data

Five sets of training data were used while building the CUED BN SUBD system, and these are specified in Table 3. The db98 data and the bn2003 were developed as STT training data sets, but, as with the CTS WW data, these sets contained punctuation marks, and therefore could be mapped to approximate the V6.2 EARS MDE annotation specification for the SUB task [8]. The basic mapping was as follows: all fullstops and commas were mapped to statement SU boundaries, while all question marks were mapped to question SU boundaries.

Name	**bnrt04**	**bnrt04_v1.0**	**bnrt03**	**db98**	**bn2003**
Epoch	2004	2004	2003	1998	2003
Released	07/09/04	04/06/04	2003	1999	2004
Spec	V6.2 (v1.1)	V6.2 (v1.0)	V5	mapped	mapped
Hours	c.20	c.20	c.20	c.90	c.4000
Tasks	all SMD	all SMD	all SMD	SUBD	SUBD

Table 3. Summary of BN training data sets used for SMD system development

4.4. BN Development Data

Three development data sets were used during the process of BN SMD system development, and these are specified in Table4

Name	**bndev03**	**bneval03**	**bndev04**
Epoch	2003	2003	2004
Spec	V6.2 (v1.1)	V6.2 (v1.1)	V6.2 (v1.1)
Hours	c.1.5	c.1.5	c.3

Table 4. Summary of BN dev data sets used for SMD system development

The bndev03 and bneval03 data sets constituted the development and evaluation sets for the RT-03f evaluation [2], while the bndev04 data set was prepared as the development set for the RT-04 evaluation.

5. PROSODIC FEATURES

Since the CUED SMD systems submitted for the RT-04 evaluation utilised audio data in order to produce output files that contained information about SMD events, prosodic features were extracted for the training and development sets. The stages in the prosodic feature extraction process were the same for all the CUED SMD systems, and these stages are detailed below.

5.1. Training Data

Forced alignments were obtained for the non-mapped CTS and BN training data sets specified in Table 1 and Table 3. The segmented files in the ctsrt04, ctsrt04_v1.0, and ctsrt03 training data sets were aligned using non-VTLN HLDA MPE triphone models. The segmented files in the bnrt04, bnrt04_v1.0, and bnrt03 training data sets were aligned using non-VTLN HLDA MPE gender dependent triphones.[5] The forced alignments obtained for both the CTS and BN training data sets provided word sequences and word-level start and end times, and, using this timing information, a set of 10 prosodic features was extracted for each lexeme token in the training data sets. The 10 prosodic features used are given in Table 5.

Prosodic Feature	**Description**
Pause_Length	pause length at end of word
Duration	duration from previous pause
Avg_F0_L	mean of good F0 values in left window
Avg_F0_R	mean of good F0 values in right window
Avg_F0_ratio	Avg_F0_L / Avg_F0_R
Cnt_F0_L	number of good F0s in left window
Cnt_F0_R	number of good F0s in right window
Eng_L	RMS energy in left window
Eng_R	RMS energy in right window
Eng_ratio	Eng_L / Eng_R

Table 5. Prosodic Features: ‘good’ F0s values are those that fall between 50Hz and 400Hz

The prosodic features in Table 5 were extracted either from the waveform data or from corresponding plp encoded data files using ESPS tools (e.g., get_f0) and CUED-internal tools [5] [6] [12]. The features were extracted from 0.2 sec window at the end of each word, and the feature vectors obtained were used in order to construct the task-specific PFMs. Since the training data for the various SMD tasks was dominated by non-SMD-event feature vectors, the set of vectors extracted for each SMD task was downsampled so that the number of non-SMD-event vectors was equal to the number of SMD-event vectors, creating a ‘50-50’ downsampled data set. These downsampled data sets were used to create the task-specific CART-style decision tree PFMs.

5.2. Dev Data

For the speech condition evaluation task, the CUED STT RT-04 20xRT output (for both the CTS and BN domains) was used as input for the CUED SMD systems, and the STT output files provided segmented hypothesised word sequences (with a start time, duration information, and confidence measures for each lexeme token) and hypothesised speaker information for each input speech signal file [13] [14]. In addition, the CTS STT output files provided channel information, indicating whether the hypothesised lexemes are associated with the speaker on channel 1 or the speaker on channel 2. Given the word sequences and word times provided by the STT output files, the 10 prosodic features specified in Table 5 were extracted for each lexeme token in the development data audio files. The task-specific PFMs output probability streams for the input feature vectors sequences obtained for the development data sets.

6. SMD SYSTEMS ARCHITECTURE

6.1. CTS and BN SUBD System Architecture

The CUED CTS SUBD system used the CUED CTS 20xRT STT output as input [13], and it used Slash Unit Language Models (SULMs) in order to detect the SU boundaries in the input files.

[5] NB: No prosodic features were extracted from the CTS WW and the BN db98 and bn2003 training data sets; these data sets were only used to train the language models for the CTS and BN SUBD task respectively, as detailed in Sections 7.1 and 7.2.

Trigram (tg) and fourgram (fg) word-based SULMs, and class-based trigram SULMs with 40 classes (cl40-tg), were constructed using each of the training data sets given in Table 1. The training data was converted into standard language model training texts, and unique tokens for the SU boundary subtypes were inserted after those lexemes that preceded the SU boundaries. SU tokens were only inserted in the boundary locations, and no special tokens were inserted after lexemes that did not constitute an SU boundary [12]. The word-based SULMs were constructed using Kneser-Ney discounting as implemented in the SRI LM Toolkit [15] [16], while the class-based SULMs were built using the HTK LM Tools [9]. The class-based N-gram SULMs were estimated, and the class-based models were trained using 4 iterations of Cluster [17].

Prosodic features were extracted for the training data as detailed in Section 5, and each feature vectors obtained was classified either as an SU feature vector (i.e., the lexeme associated with the vector constitutes an SU boundary), or else as a non-SU feature vector (i.e., the lexeme associated with the vector does not constitute an SU boundary). A separate free-standing Cart-style decision tree which functioned as a PFM was constructed for each of the training data sets. In order to compensate for the fact that c.90% of the training data consisted of non-SU vectors, 50-50 downsampled PFMs were constructed.

The SULMs were combined with the PFM in a lattice-based 1-Best Viterbi decoding framework (with the grammar scale factor set to 1). The probabilities obtained from the PFMs for each token in the dev sets were divided by their priors, (and averaged if PFMs for different training sets were combined), and the resulting likelihoods were placed on the arcs of the initial lattices which were then expanded using the SULMs and HTK lattice tools [9]. The 1-Best decoder output produced token sequences for each file in the dev sets specified in Table 2, and these contained the STT lexeme token sequence and SU boundary tokens that had been inserted automatically during the decoding process. The decoder output files were subsequently converted into scorable rttm files.[6]

The CUED BN SUBD system used the CUED BN 10xRT STT output as input [14], [18], and both word-based and class-based SULMs were built using the training data specified in Table 3. Apart from this, the CUED BN SUBD system was identical to the CUED CTS SUBD system, with one exception: for the BN SUBD system, the SULM training files and initial lattices contained special tokens indicating the segment boundaries.

6.2. CTS FWD System Architecture

The CUED CTS FWD system used the CUED CTS 20xRT STT output as input [13], and it used Filler Word Language Models (FWLMs) in order to detect the filler words in the input files. Trigram (tg) and fourgram (fg) word-based FWLMs, and class-based trigram FWLMs with 40 classes (cl40-tg), were constructed using each of the training data sets given in Table 1. The FWLMs were constructed in the same way as the SULMs (as detailed in Section 6.1).

Prosodic features were extracted for the training data as detailed in Section 5, and the feature vectors obtained were each classified either as a filler subtype, or else as a non-filler feature vector. The FWD PFM was built in the same was as the SUBD PFM (as described in Section 6.1).

The FWLMs were combined with the PFMs in a lattice-based 1-Best Viterbi decoding framework, as detailed in Section 6.1. The decoder output files were subsequently converted into scorable rttm files. Although PFMs were used in development experiments, as demonstrated in Section 7.3, the FWD PFM degraded the performance of a system that used FWLMs only.

6.3. CTS IPD System Architecture

The CUED CTS IPD system used the CUED CTS 20xRT STT output as input [13], and it used Interruption Point Language Models (IPLMs) in order to detect the interruption points in the input files. Trigram (tg) and fourgram (fg) word-based IPLMs, and class-based trigram IPLMs with 40 classes (cl40-tg), were constructed using each of the training data sets given in Table 1. Once again, these were constructed in the same manner as the SULMs and FWLMs.

Prosodic features were extracted for the training data as detailed in Section 5, and the feature vectors obtained were each classified either as IP vectors (i.e., the lexeme associated with the vector precedes an IP event), or else as a non-IP feature vector (i.e., the lexeme associated with the vector does not precede an IP event). The IPD PFMs were built as described previously.

The IPLMs were combined with the PFMs in the same lattice-based 1-Best Viterbi decoding framework that had been used for both the SUBD and FWD tasks.

7. DEVELOPMENT EXPERIMENTS FOR CUED SMD SYSTEMS

7.1. CTS SUBD System Development

The ctsrt04 training data was used to construct a single free-standing 50-50 downsampled PFM (ctsrt04_PFM) as described in Section 5. The PFM made use of 9 prosodic features and it contained ~100 terminal nodes. The average Residual Mean Deviance (RMD) for the dev sets was 1.69, and the average Misclassification Rate (MR) for the dev sets was 0.34. The ctsrt04_PFM was combined with the various SULMs as discussed below.

During the process of CTS SUBD system development, various combinations of SULMs were explored. As mentioned in Section 6.1, independent word-based and class-based SULMs were constructed for the various training data sets, and these SULMs were then interpolated, with the weights being determined manually.[7] As discussed in Section 4.1, the ctsrt04 training data was specifically prepared for the RT-04 evaluation, and it was annotated in accordance with V6.2 of the MDE Annotation Specification [8]. By contrast, the WW training data was prepared CUED-internally by mapping c.1800 hours of the WW RT-04 STT training data so that it crudely approximated the ctsrt04 data. The WW data was only used to train the SULMs, and it was not incorporated into the ctsrt04_PFM. The main results for the ctsrt04_PFM and various ctsrt04_PFM+SULM systems are given in Table6.

The WW SULMs were created in order to reduce the DEL error rate that was obtained using the ctsrt04 trained SULMs, and certainly the WW_fg achieves DEL rates that are lower than those for the ctsrt04_fg (the WW_fg DEL rates are lower by 1.9% abs, 1.0% abs, and 0.4% abs for the dev03, eval03, and dev04 sets respectively). However, as expected, the lower DEL rates are obtained at the expense of higher INS rates when performance of the WW_fg is compared to that of the ctsrt04_fg (the WW_fg INS

[6] The 'rttm' file format is defined in [7].

[7] Automated interpolation schemes proved to be suboptimal, and systems that used manually selected weights achieved lower ERR rates.

SYSTEM	%Err (DEL/INS/ERR)		
	dev03	eval03	dev04
ctsrt04_PFM	32.6/69.3/131.5	34.2/64.7/132.4	30.1/68.2/131.2
ctsrt04_fg	31.8/15.1/57.9	31.5/14.0/56.8	29.2/15.7/56.2
ctsrt04_cl40-tg	33.1/20.3/63.9	33.3/18.7/62.6	30.8/19.7/61.9
WW_fg	29.9/46.3/91.3	30.5/46.4/91.8	28.8/47.6/91.1
ctsrt04_fg+cl40-tg	31.8/14.8/57.0	31.3/13.8/56.1	29.1/14.7/54.4
ctssu_interp	30.7/15.4/**56.7**	30.4/14.3/**55.8**	28.1/15.3/**54.2**

Table 6. CTS PFM and PFM+SULM Results; all results were obtained using mdeval-v19 with the options '-w -W -t 1.00' set; the interpolated SULM ctssu_interp = ctsrt04_fg+ctsrt04_cl40-tg+WW_fg; all SULM results are for PFM+SULM systems, using the ctsrt04_PFM

rates are higher by 31.2% abs, 32.4% abs, and 31.9% abs for the dev03, eval03, and dev04 sets respectively). These comparatively high INS rates are undesirable, and they suggest that a more subtle mapping strategy would make the mapped WW data more useful.

The results in Table 6 indicate that a system that uses an interpolated ctsrt04_fg and ctsrt04_cl40-tg achieves lower ERR rates than a system that uses either of these SULMs separately, and that, despite the comparatively high WW_fg INS rates, further reductions (0.3% abs and 0.3% abs and 0.2% abs for the dev03, eval03 and dev04 sets respectively) can be achieved by interpolating the WW_fg with the ctsrt04 SULMs. The interpolation weights used in the ctssu_interp system were determined manually, and they are given in in Table 7.

SYSTEM	Interpolation Weights
ctsrt04_fg	0.575
ctsrt04_cl40-tg	0.375
WW_fg	0.050

Table 7. CTS SULM Interpolation Weights for the SULMs in the interpolated ctssu_interp SULMs

Table 7 indicates that the WW_fg SULM is given a weight that is an order of magnitude lower than those assigned to the ctsrt04 fg and cl40-tg SULMs. The ctsrt04_fg is given the highest weight, and therefore dominates the interpolated ctssu_interp SULM. When the WW_fg was assigned a higher weight (e.g., a value between 0.05 and 1.0), as expected, the DEL error rates decreased slightly, but these gains were lost because the number of INS errors increased at a faster rate, thus degrading the performance of the system.

7.2. BN SUBD System Development

The bnrt04 training data was used to build a single free-standing 50-50 downsampled PFM (bnrt04_PFM) as described in Section 5. The PFM made use of 5 prosodic features and it contained ˜60 terminal nodes. The average RMD for the dev sets was 0.96, and the average MR for the dev sets was 0.17. The bnrt04_PFM was combined with various SULMs as detailed below.

Various combinations of SULMs were explored. As mentioned in Section 6.1, independent word-based and class-based SULMs were constructed for the training data sets, and these SULMs were then interpolated, with the weights being determined manually. As discussed in Section 4.3, the bnrt04_v1.0 bnrt04 training data were specifically prepared for the RT-04 evaluation, and they were annotated using different versions of the MDE annotation specification respectively [8]. The bnrt03 training data had been prepared for the RT-03f evaluation, and had been annotated in using V5 of the MDE annotation specification [19]. By contrast, the db98 and bn2003 STT training data sets were modified CUED-internally by mapping the punctuation marks in the original transcripts so that the resulting mapped data crudely approximated the bnrt04 data. The db98 and bn2003 data were only used to train the SULMs, and they were not incorporated into the bnrt04_PFM. The main results for the bnrt04_PFM and various bnrt04_PFM+SULMs systems are given in Table8.

SYSTEM	%Err (DEL/INS/ERR)		
	dev03	eval03	dev04
bnrt04_PFM	45.2/40.2/110.2	47.3/42.2/107.9	52.0/49.1/134.0
bnrt03_tg	45.8/17.1/66.1	44.9/20.1/68.8	51.7/24.8/79.8
bnrt04_v1.0_tg	49.7/15.4/68.6	50.2/15.0/68.5	56.7/19.2/79.8
bnrt04_tg	50.4/16.0/69.9	49.4/17.2/70.2	55.9/19.9/79.0
bnrt03_cl40-tg	42.5/22.2/68.0	44.3/24.4/72.5	50.7/28.6/82.7
bnrt04_v1.0_cl40-tg	49.1/17.1/68.3	49.4/21.2/74.6	55.7/23.5/82.2
bnrt04_cl40-tg	50.2/17.5/69.5	45.2/20.6/69.0	56.1/25.6/84.8
db98_tg	29.6/35.4/67.9	31.4/44.2/80.6	40.9/45.1/89.4
db98_cl40-tg	28.0/42.9/74.4	30.1/52.7/87.8	39.1/52.6/95.7
bn2003_cl40-tg	37.1/26.9/67.4	42.4/30.1/76.8	48.4/36.2/88.6
EARS SULMs	46.1/14.8/63.4	45.4/15.3/63.9	53.7/21.7/78.8
+ db98 SULMs	42.4/16.6/61.7	42.9/16.7/63.1	52.0/22.4/77.9
+ bn2003 SULMs	41.0/17.2/**61.0**	42.1/16.8/**62.5**	51.5/22.8/**77.8**

Table 8. BN PFM and PFM+SULM Results; all results were obtained using mdeval-v19 with the options '-w -W -t 1.00' set; the EARS SULM consisted of interpolated bnrt03, bnrt04_v1.0, and bnrt04 tgs and cl40-tgs; all SULM results are for PFM+SULM systems, using the same bnrt04_PFM

As for the CTS WW_fg, the bn98 and bn2003 SULMs were created in order to reduce the DEL error rate that was obtained using the bnrt04_v1.0, bnrt04, and bnrt03 trained SULMs; and certainly the db98 and bn2003 tgs and cl40-tg achieve DEL rates that are lower than those for the bnrt04_v1.0, bnrt04, and bnrt03 tgs and cl40-tgs. However (also as for the CTS WW_fg), the lower DEL rates are obtained at the expense of higher INS rates, and these comparatively high INS rates are undesirable. Presumably, a more subtle mapping strategy would make the bn98 and bn2003 data more useful.

The results in Table 8 indicate that a system that uses an interpolated bnrt04_v1.0, bnrt04, and bnrt03 tgs and cl40-tgs achieves lower ERR rates than a system that uses these SULMs separately. Also, the results indicate that further gains (2.4% abs, 1.4% abs, and 1.0% abs for the dev03, eval03, and dev04 sets respectively) can be obtained if the db98 and bn2003 mapped-data SULMs are added to the set of interpolated models. The interpolation weights used were determined manually, and the weights for the interpolated SULM system used in the evaluation are given in in Table 7.

Table 9 indicates that the bnrt04 tg and the bnrt04_v1.0_cl40-tg dominate the interpolated SULM, while the SULMs created using the mapped db98 and bn2003 data sets are assigned comparatively low weights.

SYSTEM	Interpolation Weights
bnrt03_tg	0.10
bnrt04_v1.0_tg	0.10
bnrt04_tg	0.30
db98_tg	0.05
bnrt03_cl40-tg	0.10
bnrt04_v1.0_cl40-tg	0.20
bnrt04_cl40-tg	0.05
bn2003_cl40-tg	0.05
db98_cl40-tg	0.05

Table 9. BN SULM Interpolation Weights

7.3. CTS FWD System Development

The ctsrt04 training data was used to construct a single free-standing 50-50 downsampled PFM (ctsrt04_PFM) as described in Section 5. The PFM made use of 10 prosodic features and it contained ~100 terminal nodes. The average RMD for the dev sets was 1.58, and the average MR for the dev sets was 0.32. The ctsrt04_PFM was combined with the various FWLMs, although the FWD ctsrt04_PFM always degraded the performance of the FWD system, as discussed below.

Several combinations of FWLMs were explored. As mentioned in Section 2.2, independent word-based and class-based FWLMs were constructed for the training data sets, and these FWLMs were then interpolated, with the weights being determined manually. Some of the results for the FWLMs are given in Table 10. The best results on the dev data were obtained using word-based tg FWLMs constructed using the ctsrt03 and ctsrt04 training data sets.

SYSTEM	%Err (DEL/SUB/ERR)		
	dev03	eval03	dev04
ctsrt03_tg	35.7/12.4/49.0	36.6/12.8/50.1	31.6/9.7/41.6
ctsrt04_tg	30.0/14.8/**45.9**	32.6/16.4/49.8	26.7/11.9/39.0
ctsrt03_cl40-tg	45.5/12.8/59.1	46.3/13.9/60.1	41.5/10.8/52.8
ctsrt04_cl40-tg	41.0/14.3/55.8	41.2/16.6/58.3	36.4/13.6/50.2
fw_interp	31.8/13.8/46.4	33.7/14.6/**49.2**	27.7/10.8/**38.9**
+ ctsrt04_PFM	33.4/18.8/52.2	36.0/19.2/55.2	30.2/14.1/44.3

Table 10. CTS FWLM and FWLM+PFM Results; all results were obtained using mdeval-v19 with the options '-w -W -t 1.00' set; the fw_interp FWLM consisted of interpolated ctsrt03 and ctsrt04 tgs and cl40-tgs; the results are all for FWLM systems with no PFM, except for the fw_interp + ctsrt04_PFM system which uses the FWD ctsrt04_PFM

The results in Table 10 indicate that, for the eval03 and dev04 sets, interpolated ctsrt03 and ctsrt04 tgs and cl40-tgs achieve lower ERR rates than any of these word-based and class-based models used independently. When a FWD ctsrt04_PFM was incorporated into the decoding framework, the performance of the system degraded for all three dev sets. Specifically, the fw_interp+ctsrt04_PFM system achieves ERR rates for the dev03 eval03 and dev04 data sets that are 5.8% abs, 6.0% abs, and 5.4% abs higher than those obtained by the fw_interp system without the ctsrt04_PFM.

7.4. CTS IPD System Development

A single free-standing 50-50 downsampled PFM (ctsrt04_PFM) was built using the ctsrt04 training data as described in Section 5. The PFM made use of 8 prosodic features and it contained ~100 terminal nodes. The average RMD for the dev sets was 1.42, and the average MR for the dev sets was 0.25. The ctsrt04_PFM was combined with the various IPLMs as discussed below.

During IPD system development, numerous combinations of IPLMs were explored. As for the SULMs and FWLMs, independent word-based and class-based IPLMs were constructed for the training data sets, and these IPLMs were then interpolated, with the weights being determined manually. Some results for the IPLMs are given in Table 11. The best results on the dev data were obtained using word-based tg IPLMs constructed using the ctsrt03 and ctsrt04 data sets.

SYSTEM	%Err (DEL/SUB/ERR)		
	dev03	eval03	dev04
ctsrt03_tg	51.6/12.5/64.2	53.0/11.9/65.0	49.6/11.6/61.2
ctsrt04_tg	45.7/16.0/61.7	48.0/14.8/62.8	43.6/14.7/58.2
ctsrt03_cl40-tg	52.0/19.6/71.6	55.3/22.0/77.3	53.9/22.4/76.3
ctsrt04_cl40-tg	52.9/20.2/73.0	53.2/17.5/70.7	49.6/17.9/67.5
ip_interp	49.3/12.3/61.5	51.3/11.4/62.7	47.1/11.4/58.5
+ ctsrt04_PFM	45.7/15.7/**61.4**	48.5/13.7/**62.2**	43.9/14.2/**58.1**

Table 11. CTS IPLM and IPLM+PFM Results; all results were obtained using mdeval-v19 with the options '-w -W -t 1.00' set; the ip_interp SULM consisted of interpolated ctsrt03 and ctsrt04 tgs and cl40-tgs; the results are all for IPLM systems with no PFM, except for the ip_interp + ctsrt04_PFM system which uses the IPD ctsrt04_PFM

The results indicate that small gains can be obtained by IPLM interpolation, although the interpolated IPLMs only perform slightly better than the ctsrt04_tg (0.2% abs, and 0.6% abs respectively for the dev03, eval03 data sets; the interpolated ip_interp IPLM perform 0.3% abs worse on dev04 set than the ctsrt04_tg). When the ctsrt04_PFM is added, the ERR rates fall by about 0.4% abs on average. The interpolation weights for the IPLMs used in the ip_interp model are given in Table 12.

SYSTEM	Interpolation Weights
ctsrt03_tg	0.30
ctsrt04_tg	0.40
ctsrt03_cl40-tg	0.10
ctsrt04_cl40-tg	0.20

Table 12. CTS IPLM Interpolation Weights for the IPLMs in the interpolated ip_interp IPLM

The numbers in Table 12 indicate that the ctsrt03 tg was assigned the highest weight, and therefore dominated the interpolated IPLM. However, the ctsrt03 cl40-tg was assigned the lowest weight, which is appropriate since these models produce higher ERR rates when they are used independently.

8. RESULTS ON THE RT-04 EVALUATION DATA

8.1. RT-04 Evaluation Data

Information concerning the CTS and BN RT-04 evaluation data is given in Table 13.

Name	**ctseval04**	**bneval04**
Epoch	2004	2004
Spec	V6.2 (v1.1)	V6.2 (v1.1)
Hours	c.3	c.3

Table 13. Summary of CTS and BN RT-04 Evaluation data

This data was processed in exactly the same manner as the dev data sets had been processed. For details, see Sections 5 and 6.

8.2. CTS RT-04 Evaluation Results

The CUED CTS SMD evaluation system results for the dev sets and the RT-04 eval set are given in Table 14. These results are for both the speech and reference conditions. The reference condition involved ‘perfect’ token sequences being used as input to the SMD systems. Consequently, the reference results indicate how the various system would have performed if the STT output files used as input to the SMD systems were entirely free from errors.

SYSTEM	**%Err**			
	dev03	**eval03**	**dev04**	**eval04**
CTS FWD (spch)	52.2	55.2	44.3	45.8
CTS FWD (ref)	25.3	25.4	25.5	27.4
CTS IPD (spch)	61.4	62.2	58.1	63.5
CTS IPD (ref)	42.8	42.1	44.5	47.2
CTS SUBD (spch)	56.7	55.8	54.2	56.5
CTS SUBD (ref)	52.0	50.6	45.2	46.2

Table 14. CTS RT-04 Results

These results indicate that the SMD systems performed broadly as expected given the dev set results. Since the CTS FWD and CTS IPD systems were constructed for RT-04, the main progress since RT-03f in relation to these systems is that they now form a baseline for future research. A CTS SUBD system was submitted for RT-03f, but determining progress since that evaluation is not trivial since the task definition has changed: for RT-03f SUB errors were not scored, while these errors were scored for RT-04. Consequently, the RT-03f SUBD system used a posterior decoding scheme that did not take SU subtypes into account [6]. However, the results in Table 15 provide a comparison of the RT-03f and RT-04 CTS SUBD evaluation system, by presenting numbers for the RT-03f and RT-04 system output files when scored against versions of the eval03 reference files that are annotated using V5 and V6.2 of the MDE annotation specification respectively.

The results in Table 15 suggest that (ignoring SUB errors) the RT-04 system achieves ERR rates that are between c.5% and c.11% abs lower than those achieved by the RT-03f system.

8.3. BN RT-04 Evaluation Results

The CUED BN SMD evaluation system results for the dev sets and the RT-04 eval set are given in Table 16. Once again, the results presented are for both the speech and reference conditions where the reference condition involved ‘perfect’ token sequences being used as input to the SMD systems. Consequently, the reference results indicate how the various system would have performed if the STT output files used as input to the SMD systems were entirely free from errors.

SYSTEM	**DEL**	**INS**	**SUBS**	**%Err (DEL/INS)**
RT-03f_sys/V5_ref	33.1	19.3	11.7	64.1 (52.4)
RT-03f_sys/V6.2_ref	34.1	21.2	10.9	66.1 (55.2)
RT-04_sys/V5_ref	32.0	15.1	13.9	61.0 (47.1)
RT-04_sys/V6.2_ref	30.4	14.3	11.2	55.8 (44.7)

Table 15. Results for RT-03 and RT-04 eval systems using V5 and V6 ref files for eval03 data

SYSTEM	**%Err**			
	dev03	**eval03**	**dev04**	**eval04**
BN SUBD (spch)	61.0	62.5	77.8	72.2
BN SUBD (ref)	57.5	60.6	75.1	71.1

Table 16. BN RT-04 Results

These results indicate that the dev04 and eval04 data sets were more difficult than the dev03 and eval03 sets, and this roughly corresponds to STT performance on these data sets [14] [18].

9. CONCLUSIONS

This paper has summarised the CUED SMD systems that were developed as part of the EARS RT-04 evaluation. Since all except one of these systems was developed from scratch specifically for the RT-04 evaluation, the systems described in this paper primarily constitute an initial attempt to solve the non-trivial CTS FWD, CTS IPD, and BN SUBD pattern recognition problems defined within the EARS SMD framework. In general, these results suggest that is is possible to use the same generic system architecture for the FWD, IPD, and SUBD tasks, adapting the same general framework for each specific task. For instance, particular framework modifications can include the use of PFMs and task-specific variations in the number and types of language models. The results presented here suggest that the FWD task is best approached as a pattern recognition problem that does not utilise acoustic information as an independent information source once the STT stage in the process is complete. By contrast, the IPD system achieve the lowest ERR rates when the interpolated IPLM are combined with IP-based PFMs. The RT-04 CTS SUBD system gave an c.8% abs improvement over the RT-03f system, and the CTS and BN SUBD results in general suggest that these systems may benefit by using larger amounts of mapped training data. Since small gains are obtained for both the CTS and BN SUBD tasks using SULM trained on crudely mapped data, it is reasonable to assume that a more subtle automated punctuation-to-SU mapping could enable the large STT training data sets (which contain thousands rather than tens of hours of data) to be utilised as training data by the SMD SUBD systems. If such a mapping were developed, then the amount of available training data for the SUBD tasks would increase by several orders of magnitude.

10. REFERENCES

[1] NIST, "Benchmark Tests : Rich Transcription (RT) ," `http://www.nist.gov/speech/tests/rt/`.

[2] NIST, "The Rich Transcription Fall 2003 (RT-03F) Evaluation Plan, version 4," `http://www.nist.gov/speech/tests/rt/rt2003/fall/docs/rt03-fall-eval-plan-v9.pdf`, 9th October 2003.

[3] T. Gibson and D. Jones, "Psycholinguistic Experiments Measuring the Effects of MDE on Readability," Proc. Fall 2003 Rich Transcription Workshop (RT-03f), 2003.

[4] C. Wayne, "TIDES Mission Statement," `http://www.darpa.mil/ipto/programs/tides/index.htm`, 2004.

[5] M. Tomalin and P. C. Woodland, "Structural Metadata at CUED: Progress Report," in *EARS Workshop May 21st*. 2003, Boston MA.

[6] M. Tomalin and P. C. Woodland, "SU Detection for RT-03f at Cambridge University," in *RT-03f Workshop 13th November*. 2003, Washington D.C.

[7] NIST, "Fall 2004 Rich Transcription (RT-04F) Evaluation Plan," `http://www.nist.gov/speech/tests/rt/rt2004/fall/docs/rt04f-eval-plan-v14.pdf`, 30th August 2004.

[8] Linguistic Data Consortium, "Simple Metadata Annotation Specification V6.2," `http://www.nist.gov/speech/tests/rt/rt2004/fall/docs/rt04f-eval-plan-v14.pdf`, 2004.

[9] S. J. Young, G. Evermann, M. J. F. Gales, T. Hain, X. L. Liu, J. J. Odell, D. Ollason, D. Povey, V. Valtchev, and P. C. Woodland, "The HTK Book," `http://htk.eng.cam.ac.uk`, 2003.

[10] J-H. Kim, "Named entity recognition from speech and its use in the generation of enhanced speech recognition output," PhD Thesis, CUED, 2001.

[11] J-H. Kim and Woodland P. C., "The Use of Prosody in a Combined System for Punctuation Generation and Speech Recognition," in *Proceedings of the European Conference on Speech Communication and Technology*, 2001, pp. 2757–2760.

[12] M. Tomalin and P. C. Woodland, "Advances in Structural Metadata at CUED," MDE Technical Meeting, Washington DC, 1st May, 2004.

[13] G. Evermann, H.Y. Chan, M.J.F. Gales, B. Jia, D. Mrva, K.C. Sim, P.C. Woodland, and K. Yu, "Development of the 2004 CU-HTK English CTS systems using two thousands hours of data," in *Proc. Fall 2004 Rich Transcription Workshop (RT-04)*, November 2004.

[14] D. Y. Kim, H. Y. Chan, G. Evermann, M. J. F. Gales, D. Mrva, K. C. Sim, and P. C. Woodland, "Recent Developments at Cambridge in Broadcast News Transcription," in *Proc. Fall 2004 Rich Transcription Workshop (RT-04)*, November 2004.

[15] A. Stolke, "The SRI Language Modelling Toolkit," `http://www.speech.sri.com/projects/srilm`, 2004.

[16] A. Stolke, "SRILM - an extensible language modeling toolkit," in *Proc. Intl. Conf. on Spoken Language Processing*, 2002.

[17] G. Moore and S. J. Young, "Class-based Language Model Adaptation using Mixtures of Word-class Weights," in *Proc. ICSLP*, 2000.

[18] D. Y. Kim, H. Y. Chan, G. Evermann, M. J. F. Gales, D. Mrva, K. C. Sim, and P. C. Woodland, "Development of the CU-HTK 2004 Broadcast News Transcription Systems," in *Proc. ICASSP*, March 2005.

[19] Linguistic Data Consortium, "Simple Metadata Annotation Specification V5," `http://www.nist.gov/speech/tests/rt/rt2003/fall/docs/SimpleMDE_V5.0.pdf`, 2003.