The IBM 2006 Speech Transcription System for European Parliamentary Speeches

B. Ramabhadran, O. Siohan, L. Mangu, G. Zweig   M. Westphal. H. Schulz, A. Soneiro

IBM T. J. Watson Research Center
Yorktown Heights, NY

EMEA Voice Technology Development
IBM Germany

Abstract

TC-STAR is an European Union funded speech to speech translation project to transcribe, translate and synthesize European Parliamentary Plenary Speeches (EPPS). This paper describes IBM's English and Spanish speech recognition systems submitted to the TC-STAR 2006 Evaluation. The technical advances in this submission include two different algorithms for automatic segmentation and speaker clustering of the input audio; a system architecture that is based on cross-adaptation across these two segmentation schemes and system combination through generation of an ensemble of systems using randomized decision tree state-tying; automatic punctuation of the speech recognition output; and the incorporation of an additional 35 hours of in-domain EPPS acoustic training data. These advances reduced the error rate by 30% relative over the best-performing system in the TC-STAR 2005 Evaluation on the 2006 English development test set, and produced one of the best performing systems on the 2006 evaluation in English with a word error rate of 8.3%.

Index Terms: speech recognition, automatic segmentation, cross-adaptation, randomized decision trees, TC-STAR.

1. Introduction

The TC-STAR (Technology and Corpora for Speech to Speech Translation) project financed by the European Commission within the Sixth framework Program is a long-term effort to advance research in speech to speech translation technologies[1]. The primary goal of the TC-STAR project is to produce an end-to-end system in English and Spanish that accepts parliamentary speeches in one language, transcribes, translates and synthesizes them into another language, while significantly reducing the gap between the performance of a human (interpreter) and a machine. To support this goal, the performance of each component technology, namely, speech recognition (ASR), machine translation (MT) and text-to-speech (TTS) is optimized to produce the best output at their respective stages. The 2006 Evaluation was open to external participants as well as the TC-STAR partner sites [12].

The EPPS corpus comprises of over 700 politicians discussing current affairs during several public sessions of the European Parliament in multiple languages and the minutes of these sessions edited by the European parliament also known as the Final Text Editions. In the 2006 evaluation, the training, development and evaluation data comprised of recordings made between April 1996 and September 2005. Within the TC-STAR project, the evaluation is done under three different conditions:

- *public*, which allows the use of any data that is publicly available, such as Broadcast news and web data in addition to the EPPS acoustic training data and Final text Editions;
- *restricted*, which allows the use of EPPS data only; and
- *open*, which allows the use of publicly available and any inhouse material in addition to the EPPS data.

This paper describes the IBM system submitted under the *public* and *restricted* conditions. The key design characteristics for both evaluation conditions include:

- An architecture that uses two different speaker segmentation and clustering schemes and uses the output of the system using one scheme to cross adapt the same models to the second scheme;
- System combination via ROVER of multiple ASR systems built using a randomized decision-tree growing procedure [1] and cross adapted across two speaker segmentation schemes[2];
- A basic set of models that use VTLN and SAT training followed by fMPE+MPE training [2] and speaker adaptation using MLLR;
- Rescoring of the lattices produced after MLLR with an in-domain language model (restricted condition) and out-of-domain language model (public condition). This is the only step that uses non-EPPS training material;
- Static decoding graph with quinphone context;
- Training of acoustic models using EPPS material only; and
- Automatic punctuation of the final output with periods and commas in a post-processing step.

The rest of this paper is organized as follows. Section 2 describes the technical advances in this year's system; Section 3 describes the training and test material used in this task; Section 4 describes the system basics including the lexicon and acoustic and language models; Section 5 describes the overall system architecture; Section 6 describes the automatic punctuator and Section 7 presents the results.

2. Algorithms

The 2006 IBM TC-STAR speech recognition system is organized around an architecture that combines multiple systems through cross-adaptation across different segmentation schemes and ROVER. This section explains the algorithms using the Eglish system on the Dev06 test set.

2.1. Speaker Segmentation and Clustering

The EPPS rescordings contain speeches from politicians and interpreters in different languages, both from native and non-native speakers of English. In this year's evaluation, the number of speakers and their speech boundaries were not provided. Therefore, the first step in the recognition system is a segmentation of each session's audio file into speech and non-speech segments, followed by deletion of the non-speech segments. We use an HMM-based segmentation

[1] Project No. FP6-506738

[2] This is not part of the 2006 Spanish ASR system

system that models speech and non-speech segments with five-state, left-to-right HMMs with no skip states. The output distributions in each HMM are tied across all states in the HMM, and are modeled with a mixture of diagonal-covariance Gaussian densities. The speech and non-speech models are obtained by applying a likelihood-based, bottom-up clustering procedure to the speaker independent acoustic model used in the first pass of the decoding step. The speech segments are then segmented into homogeneous segments using the change-point detection procedure described in [3]. This is followed by a clustering procedure to cluster the segments into clusters that can then be used for speaker adaptation. Two different clustering procedures were used:

S1: All homogeneous speech segments are modeled using a single Gaussian density and clustered into a pre-specified number of clusters using K-means and a Mahalahobis distance measure (185 and 186 speaker clusters were identified in the Dev06 and Evl06 test sets).

S2: All homogeneous speech segments are modeled using a 4 Gaussian mixture GMM, and an iterative segmentation/reestimation procedure similar to [4] is used to define pseudo-speaker segments (110 and 132 speaker clusters were identified in the Dev06 and Evl06 test sets).

2.2. Cross segmentation adaptation

Adaptation across different segmentation and speaker clustering schemes is a form of cross-system adaptation which involves computing speaker-specific transforms from ASR transcripts generated from different systems. It aligns the transcripts generated from a single system using one speaker clustering scheme (for example, S1) to the speaker assignments generated by a second clustering scheme (for example, S2) and adapts the same acoustic models with the newly aligned transcripts. In our system, we chose transcripts from the speaker segmentation scheme with the best speaker independent performance (S1) for aligning with the speaker clusters from scheme, S2. A comparison of error rates between self and cross adaptation studied on the Dev06 test set using one set of acoustic models is shown in Table 1. A gain of 0.8% absolute is obtained with cross adaptation when compared to 0.3% to 0.4% absolute seen with self adaptation.

	SI	MLLR
S1	12.7	12.3
S2	13.0	12.8
S1.SI→S2.MLLR	-	11.9

Table 1: Comparison of WER: Effect of cross segmentation adaptation on Dev06 test set.

2.3. Ensemble of ASR systems using randomized decision trees

A characteristic of our system architecture is the use of an ensemble of ASR systems whose decisions are combined using ROVER [5] to obtain a single recognition hypothesis (see Fig. 1). The ROVER voting approach is most effective when the individual ASR systems of the ensemble make uncorrelated errors. A typical procedure to build such systems is to use different acoustic front-ends (e.g. PLP vs MFCC) or different phone sets across systems. In this work however, we adopt a more systematic approach to build multiple systems by randomizing the training procedure. Randomness is introduced by replacing the classical decision-tree state-tying procedure used to tie context-dependent acoustic units, by a randomized decision tree growing procedure [1]. Randomized decision trees are grown by randomly selecting the split at each node, from the top N-best split candidates. In contrast, standard decision trees are grown by selecting the best split candidate. ASR systems built on different sets of randomized decision trees will model different clusters of context-dependent units. Multiple systems can then be systematically built simply by changing the random number generator seed. We have experimentally observed that such systems are good candidates to be used with the ROVER voting procedure [1]. Table 2 demonstrates a 0.6% reduction in WER on the Dev06 test set where the top 5 candidates were considered for the split. Three different systems were built with 6000 states and 150K Gaussians and combined with the baseline system of the same size.

	Baseline	R1	R2	R3	ROVER
WER	11.0	11.2	11.4	11.1	10.4

Table 2: Comparison of WER: Effect of using an ensemble of ASR systems on the Dev06 test set.

3. Training Data

The English training data comprises of 101 hours of the English portion of the EU plenary sessions with approximately 75 hours of speech from over 1900 speakers (politicians and interpreters) . This data covers sessions from May 2004 through May 2005. The Dev06 development test set on which the acoustic and language models were optimized consists of approximately 3 hours of data from 42 speakers (mostly non-native speakers). The 2006 English Evaluation corpus comprises of 3 hours of data from 41 speakers and the Spanish corpus consists of 3 hours each from EPPS and Spanish Parliament speeches. The text sources for language model training include the acoustic training transcripts (755K words), the final text editions (37M words), web data released by the University of Washington (525M words) and Broadcast News data (204M words). The Spanish acoustic training material had approximately 95 hours of EPPS data and 33M words from FTE, with an additional 43M words of transcribed text from the Spanish Parliament available for language model training.

4. Basic System Description

4.1. Acoustic Modeling

The acoustic front-end employs 40-dimensional, perceptual linear prediction (PLP) features obtained from an LDA projection that are mean and variance normalized on a per utterance basis.

The speaker-independent (SI) acoustic models used in the system consists of multiple sets of HMMs all of them trained on all transcribed EPPS acoustic material available for training as released by RWTH for this project. The speaker-independent model uses continuous density left-to-right HMMs using Gaussian mixture emission distributions and uniform transition probabilities. The number of mixtures for a tied state s with C_s observations is given by $4C_s^{0.2}$. A global Semi-Tied Covariance (STC) [6, 7] linear transformation is used. Each HMM has 3 states except for the silence HMM which is a single state model. The English system uses 45 phones, 42 speech phones, 1 silence phone and 2 noise phones. The speech HMMs use 3172 context dependent quinphone states modeled by 95K Gaussians. The Spanish system uses 49 speech phones and 4 noise phones modeled using 4000 states and 100K Gaussians.

The evaluation system employs Vocal Tract Length Normalization (VTLN) [8, 9]. The frequency warping is piecewise linear using a breakpoint at 6500Hz. It is estimated from among 21 candidate warping factors ranging from 0.8 to 1.2 in steps of 0.02 using a full-covariance, voicing model built from 13-dimensional PLP features.

The VTLN model is trained on features in the warped space, using an LDA transformation and decision tree clustering of quinphone statistics to yield 6000 tied-states and 100K Gaussians for both English and Spanish.

The SAT model [6, 7] is trained on features in a linearly transformed feature space resulting from applying fMLLR transforms computed on a per speaker basis to the VTLN normalized features. Several sets of SAT HMMs were built for English using decision tree clustering of quinphone statistics:

Model A: 6000 tied-states, 150K Gaussian system

Model B: 8000 tied-states, 180K Gaussian system

Model R1,R2 and R3: Three 6000 tied-states, 150K Gaussian systems obtained by using randomized decision tree growing procedure described in Section 2.3

For Spanish, the SAT model had 6000 states modeled by 100K Gaussians.

The MPE model is trained on features obtained from a feature-space minimum phone error (fMPE) transformation [2]. The fMPE projection uses 1024 Gaussians obtained from clustering the Gaussian components in the SAT model. Posterior probabilities are then computed for these Gaussians for each frame. The fMPE transformation maps the high dimensional posterior-based observation space to a 40-dimensional fMPE feature space. The MPE model is then trained in this feature space using 3 iterations of training using a Minimum Phone Error criterion described in [10].

4.2. Language Modeling

4.2.1. English

All decoding passes use a 4-gram modified Knesser-Ney model that was built using the SRI LM toolkit [11] using the various sources described in Section 3. One model was trained on the training transcripts (LM1) and another on the text corpus based on the Final Text Editions (LM2). A perplexity minimizing mixing factor was computed using the Dev06 reference text. The final interpolated language model used in the construction of the static decoding graph contains 5.5M ngrams. The interpolation weights assigned to the out-of-domain language models LM3 and LM4 is relatively low, 0.12 and 0.13 compared to 0.21 and 0.54 for LM1 and LM2. For the language model rescoring step, two different interpolated language models corresponding to the *restricted* and *public* conditions were built. For the *public* condition, two additional models were trained on out-of-domain text sources and interpolated with LM1 and LM2. LM3 containing 80M n-grams was trained on 525M words of web data released by the University of Washington and LM4 containing 39M n-grams was built on 204M words of Broadcast News. The final interpolated LM contains 130M ngrams. The lattice rescoring step for the *restricted* condition used a larger interpolated LM (9M ngrams from LM1 and LM2) built from the EPPS sources.

The 59K recognition lexicon was obtained by taking all words occurring at least twice in the text corpus and once in the the acoustic training transcripts. The OOV rate on the dev06 test set was slightly under 0.4%. This language model, lexicon and HMM components were then used to build a static decoding graph of 57M states and 248M arcs.

4.2.2. Spanish

The language model for the decoding passes, built from the EPPS acoustic training text, the FTE text and the Spanish Parliament text contained 8M ngrams, while a larger model containing 90M ngrams was used for the rescoring pass. The 65k recognition lexicon was obtained by taking all words occurring at least 6 times in the EU and 8 times in the Spanish parliament text corpus and once in the acoustic training transcripts. Names of all members of the EU parliament were added. The OOV rate on the Dev06 test set was 1.2%.

5. Overall System Architecture

It is common in evaluation systems to combine recognition systems that make complimentary errors to produce the final output. This section describes the overall system architecture detailing the decoding steps and acoustic models (described in Section 4.1) that include the algorithms described in Section 2.

The first step is speech segmentation and speaker clustering where the speaker clusters corresponding to the two schemes S1 and S2 are determined. This is followed by decoding steps (a) through (f) described below and represented by a single block (labeled as “Baseline”) in Figure 1. Model A was used to decode the test data in 6 passes using segmentation scheme S1.

a) The SI pass uses the SI model and the LDA projected PLP features.

b) Using the transcript from a) as supervision, warp factors are estimated for each cluster using the voicing model and a new transcript is obtained by decoding using the VTLN model and VTLN warped features.

c) Using the transcript from b) as supervision, fMLLR transforms are estimated for each speaker cluster using the SAT model. A new transcript is obtained by decoding using the SAT model and the fMLLR transformed VTLN features.

d) The VTLN features after applying the fMLLR transforms are subjected to the fMPE transform and a new transcript is obtained by decoding using the MPE model and the fMPE features.

e) Using the transcript from d) as supervision, MLLR transforms are estimated for each cluster using the MPE model.

f) The lattices resulting from e) are rescored using the 4-way interpolated language model described in 4.2. The one-best at this step will be referred to as CTM.

Model B was used to decode the test data using segmentation scheme S2 from step (a) through step (c), i.e., to obtain the vtln warp factors and the fMLLR transforms corresponding to the speaker clusters in S2. For cross-segmentation adaptation, CTM (from step (f) above) is now used as the reference transcript to compute MLLR transforms for Model B and process steps (e) and (f) using S2. The one-best from this stage will be referred to as CTM′. The above steps (a) through (f) are applied to the three different models built using randomized decision trees (R1, R2 and R3) using segmentation scheme S1. These three decodes from segmentation S1 were subsequently used to cross adapt the models, R1, R2 and R3 and redecode the test data using segmentation scheme S2. Finally, three different decoded outputs CTM-R1′, CTM-R2′ and CTM-R3′ were obtained. Last, CTM′, CTM-R1′, CTM-R2′ and CTM-R3′ were rovered together to produce the final output.

6. Automatic Punctuation

Translation systems typically use punctuation marks as sentence delimiters. This motivated the development of an automatic punctuator. The input to the punctuator is the final output of the ASR system containing all word and non-word (silence, noise, laughter, breath, etc) events. Half of the English Dev06 test set was used for training and the remaining half for development. Classifiers were built only for periods and commas given the lack of “!” in the training data as well as inconsistencies in the reference set. Punctuation marks are hypothesised only in the regions of non-word events due to the high correlation observed (98% for periods and 70% for commas) between the

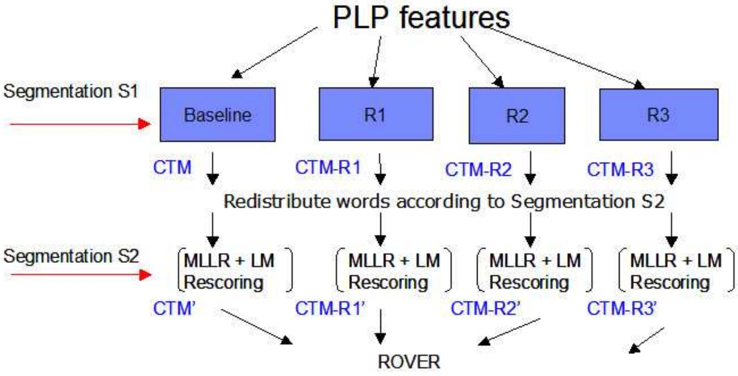

Figure 1: Overall System Architecture

two. For a given contiguous sequence of non-words a Maximum Entropy classifier is used to decide if this sequence should be a period or a comma. The identity and duration of the non-words, the difference in the LM probability between the word sequence containing the punctuation symbol in the middle and the one without, the total duration of the non-word region, and the unigram, bigram and trigram contexts surrounding the non-word region were used as input features for the maximum entropy classifier. The set of features used in the classifier was determined by optimizing the Slot Error Rate metric in the reference set. This punctuator had the best performance in the 2006 TC-STAR evaluation.

7. Results

Table 3 illustrates the gains obtained at each stage of the decoding process using one segmentation scheme and the baseline acoustic models (Model A). Discriminative training provides upto 2.7% absolute gain over the SAT system MLLR adaptation. The transcripts corresponding to the last row in Table 3 were then aligned to speakers identified using segmentation scheme S2 and the acoustic models (Model B) were adapted using the newly aligned transcripts.[3] Table 4 shows the results of cross-segmentation adaptation followed by lattice rescoring. The final numbers for the *public* and *restricted* conditions on the Dev06 and Evl06 test sets are given in Table 5. The English system uses the combination of multiple, cross-adapted ASR systems according to the procedure described in Section 5.

System	English	Spanish
SI	18.5	12.3
VTLN	18.1	10.8
SAT	15.4	9.7
fMPE+MPE	12.7	8.7
MLLR	12.3	8.7
LM Rescoring	11.6	7.8

Table 3: Error Rates of the baseline system on the Dev06 test sets for English and Spanish using segmentation scheme S1.

System	English	Spanish
Cross-seg based MLLR	11.9	8.2
LM Rescoring	11.0	7.7

Table 4: Performance improvements with cross-segmentation adaptation and rescoring on Dev06 test set.

	English		Spanish
System	Public	Restricted	Restricted
Dev 06	10.4	10.9	7.7
Evl 06	8.3	8.9	8.6

Table 5: WERs on the Dev06 and Evl06 EPPS test sets using the 2006 Evaluation System.

8. Conclusions

We have presented the IBM 2006 TC-STAR ASR system. It can be seen that adaptation across segmentation schemes voting across multiple, systematically-built ASR systems account for approximately 1% absolute reduction in WER. Additional out-of-domain training data provides a reduction in WER of only 0.5%.

9. References

[1] O. Siohan, B. Ramabhadran, and B. Kingsbury, "Constructing ensembles of ASR systems using randomized decision trees," in *Proc. IEEE ICASSP*, Philadelphia, USA, 2005.

[2] D. Povey, B. Kingsbury, L. Mangu, G. Saon, H. Soltau, and G. Zweig, "fMPE: Discriminatively trained features for speech recognition," in *Proc. IEEE ICASSP*, Philadelphia, USA, 2005.

[3] J. Ajmera and C. Wooters, "A robust speaker clustering algorithm," in *IEEE Proc ASRU Workshop*, US Virgin Islands, 2003, pp. 411–416.

[4] J. L. Gauvain, L. Lamel, and G. Adda, "The LIMSI broadcast news transcription system," *Speech Communication*, vol. 37, no. 1–2, pp. 89–108, 2002.

[5] J. G. Fiscus, "A post-processing system to yield reduced word error rates: Recogniser output voting error reduction (ROVER)," in *Proc. IEEE ASRU Workshop*, Santa Barbara, 1997, pp. 347–352.

[6] M. F. J. Gales, "Maximum likelihood linear transformations for hmm-based speech recognition," *Computer Speech and Language*, , no. 12, pp. 75–98, 1998.

[7] G. Saon, G. Zweig, and M. Padmanabhan, "Linear feature space transformations for speaker adaptation," in *Proc. IEEE ICASSP*, Salt Lake City, Utah, USA, 2001.

[8] S. Wegman, D. McAllaster, J. Orloff, and B. Peskin, "Speaker normalization on conversational telephone speech," in *Proc. IEEE ICASSP*, Atlanta, Georgia, USA, 1996.

[9] G. Saon, M. Padmanabhan, and R. Gopinath, "Eliminating inter-speaker variability prior to discriminant transforms," in *IEE Proc. ASRU Workshop*, Madonna di Campiglio, Italy, 2001.

[10] D. Povey and P. C. Woodland, "Minimum phone error and i-smoothing for improved discriminative training," in *Proc. IEEE ICASSP*, Orlando, Florida, USA, 2002.

[11] A. Stolcke, "SRILM - an extensible language modeling toolkit," in *Proc. ICSLP*, Denver, Colorado, USA, 2002.

[12] http://www.tc-star.org

[3] Use of Model A or Model B at this step does not produce a different result.