



AFRL-RH-WP-TR-2011-0027

SPEECH PROCESSING AND RECOGNITION (SPaRe)

**David M. Hoeferlin
Brian M. Ore
Stephen A. Thorn
David Snyder**

**SRA International, Inc.
5000 Springfield Street, Suite 200
Dayton OH 45431**

**JANUARY, 2011
Final Report**

Distribution A: Approved for public release; distribution is unlimited.

**AIR FORCE RESEARCH LABORATORY
711TH HUMAN PERFORMANCE WING,
HUMAN EFFECTIVENESS DIRECTORATE,
WRIGHT-PATTERSON AIR FORCE BASE, OH 45433
AIR FORCE MATERIEL COMMAND
UNITED STATES AIR FORCE**

NOTICE AND SIGNATURE PAGE

Using Government drawings, specifications, or other data included in this document for any purpose other than Government procurement does not in any way obligate the U.S. Government. The fact that the Government formulated or supplied the drawings, specifications, or other data does not license the holder or any other person or corporation; or convey any rights or permission to manufacture, use, or sell any patented invention that may relate to them.

This report was cleared for public release by the 88th Air Base Wing Public Affairs Office and is available to the general public, including foreign nationals. Copies may be obtained from the Defense Technical Information Center (DTIC) (<http://www.dtic.mil>).

AFRL-RH-WP-TR-2011-0027 HAS BEEN REVIEWED AND IS APPROVED FOR PUBLICATION IN ACCORDANCE WITH ASSIGNED DISTRIBUTION STATEMENT.

//SIGNED//
RAYMOND E. SLYH
Work Unit Manager
Sensemaking & Organizational
Effectiveness Branch

//SIGNED//
DAVID G. HAGSTROM
Anticipate & Influence Behavior Division
Human Effectiveness Directorate
711th Human Performance Wing
Air Force Research Laboratory

This report is published in the interest of scientific and technical information exchange, and its publication does not constitute the Government's approval or disapproval of its ideas or findings.

REPORT DOCUMENTATION PAGE				Form Approved OMB No. 0704-0188	
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing this collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number. PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.					
1. REPORT DATE (DD-MM-YYYY) 30-01-2011		2. REPORT TYPE Final		3. DATES COVERED (From - To) April 2009 – September 2010	
4. TITLE AND SUBTITLE Speech Processing and Recognition (SPaRe)				5a. CONTRACT NUMBER FA8650-09-D-6939 TO 0002	
				5b. GRANT NUMBER	
				5c. PROGRAM ELEMENT NUMBER 62202F	
6. AUTHOR(S) David M. Hoeflerlin, Brian M. Ore, Stephen A. Thorn, David Snyder				5d. PROJECT NUMBER 7184	
				5e. TASK NUMBER 08	
				5f. WORK UNIT NUMBER 71840871	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) SRA International, Inc. 5000 Springfield Street, Suite 200 Dayton OH 45431				8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) Air Force Materiel Command Air Force Research Laboratory 711th Human Performance Wing Human Effectiveness Directorate Anticipate & Influence Behavior Division Sensemaking & Organizational Effectiveness Branch Wright-Patterson AFB OH 45433-7022				10. SPONSOR/MONITOR'S ACRONYM(S) 711 HPW/RHXS	
				11. SPONSOR/MONITOR'S REPORT NUMBER(S) AFRL-RH-WP-TR-2011-0027	
12. DISTRIBUTION / AVAILABILITY STATEMENT Distribution A: Approved for public release; distributed is unlimited.					
13. SUPPLEMENTARY NOTES 88ABW/PA cleared on 25 March 2011, 88ABW-2011-1714.					
14. ABSTRACT This final report provides research results in the areas of automatic speech recognition (ASR), speech processing, machine translation (MT), natural language processing (NLP), and information retrieval (IR).					
15. SUBJECT TERMS Automatic Speech Recognition (ASR), Speech Processing, Machine Translation (MT), Natural Language Processing (NLP), Information Retrieval (IR)					
16. SECURITY CLASSIFICATION OF: UNCLASSIFIED			17. LIMITATION OF ABSTRACT SAR	18. NUMBER OF PAGES 60	19a. NAME OF RESPONSIBLE PERSON Raymond E. Slyh
a. REPORT U	b. ABSTRACT U	c. THIS PAGE U			19b. TELEPHONE NUMBER (include area code) NA

THIS PAGE LEFT INTENTIONALLY BLANK

TABLE OF CONTENTS

Section	Page
SUMMARY	1
1.0 INTRODUCTION	3
2.0 EXPERIMENTS AND ACCOMPLISHMENTS	4
2.1 Automatic Speech Recognition.....	4
2.1.1 ASR on the ARL Dari Corpus	4
2.1.1.1 ASR System Development	4
2.1.1.2 MLP Features	8
2.1.1.3 Data Constrained Experiments	10
2.1.2 Dari and Pashto ASR on the TRANSTAC Corpora	14
2.1.3 Mandarin Broadcast News ASR	17
2.1.3.1 Baseline ASR System.....	17
2.1.3.2 ASR Training with Approximate Transcripts	19
2.1.4 RWTH ASR.....	21
2.1.5 Sphinx-4 Recognizer.....	22
2.1.5.1 HDecode and Sphinx-4.....	22
2.1.5.2 MFCC Feature Computation	23
2.1.6 Summary	25
2.1.7 Recommendations for Future Work.....	26
2.2 Information Extraction and Retrieval.....	27
2.2.1 Architecture.....	28
2.2.2 User Interface.....	29
2.2.3 Back-End Processing	30
2.2.4 Lucene / Solr Search Index	30
2.2.5 Video Processing	32
2.2.6 English and Mandarin Gender Detection.....	33
2.2.7 Summary	33
2.2.8 Recommendations for Future Work.....	34
2.3 SCREAM Wikipedia Aided Translation.....	34
2.3.1 Initial Approach	35
2.3.2 Refined Approach	35
2.3.3 Web Interface.....	38
2.3.4 Creating (or Recreating) a SWAT Database.....	40
2.3.4.1 Create The MySQL Database.....	40
2.3.4.2 Download Wikipedia Dumps	40
2.3.4.3 Building The SWAT Database	41
2.3.4.4 Generating Additional Indices and Language Counts	42
2.3.5 Summary	42
2.3.6 Recommendations for Future Work.....	42
2.4 System Administration Support	43
3.0 CONCLUSIONS AND RECOMMENDATIONS	44
REFERENCES	46
LIST OF ACRONYMS & GLOSSARY	48

LIST OF FIGURES

Figure	Page
Figure 1: (A) Three-state Silence, (B) One-state Short-pause, and (C) Three-state Short-pause HMMs. Only States 2, 3, and 4 are Emitting, and the Arcs Show the Allowable Transitions.	6
Figure 2: ARL Dari WERs using MFCCs with Varying Numbers of Shared States and Mixtures.	7
Figure 3: ARL Dari WERs using MFCC-MLP Features with Varying Numbers of Shared States and Mixture Components.	9
Figure 4: ARL Dari WERs Obtained using Varying Amounts of Test Data to Estimate the CMLLR Transforms. All HMMs were Estimated with SAT.	11
Figure 5: ARL Dari WERs using Varying Amounts of Acoustic and LM Training Data. The Left Side of the Chart Shows the WERs for each System without SAT, and the Right Side Shows the WERs for each System with SAT.	13
Figure 6: ARL Dari WERs using Varying Amounts of Training Data. Each HMM System was Evaluated using a Data Constrained LM, and an LM Estimated on 20 Hours of Data.	14
Figure 7: Mandarin CER on each Show from the HUB4, RT-03, and RT-04 Test Sets. The same HMMs were used for each Task with the HUB4 LM and Gigaword LM.	19
Figure 8: Mandarin CER on each Show from the HUB4, RT-03, and RT-04 Test Sets. The Gigaword LM was used for all Tasks, and the HMMs were Trained on the HUB4 Corpus, and the HUB4 TDT4 Corpus.	20
Figure 9: Haystack Concept Diagram.	28
Figure 10: Haystack Architecture	29
Figure 11: Configuring the DataSource	30
Figure 12: Data-config.xml	31
Figure 13: The Metadata is also Copied into a Catchall 'Text' Field.	31
Figure 14: Y'CbCr420p Pixel Layout for each Frame. In this Example, the Resolution of the Video Frame would be 6 x 4 Pixels. Note that the Data Stream for this Video Frame would be {Y0, Y1, Y2, ..., Y23, Cb0, Cb1, Cb2, Cb3, Cb4, Cb5, Cr0, Cr1, Cr2, Cr3, Cr4, Cr5}..	32
Figure 15: Wikipedia Interlanguage Links	35
Figure 16: Wikipedia and SWAT Database Tables	36
Figure 17: SWAT Web-Based User Interface	39

LIST OF TABLES

Table	Page
Table 1: Dari WER Comparison of MFCC and MFCC-MLP Features.	10
Table 2: ARL Dari WERs Obtained using an Incremental Approach for Computing the CMLLR Transforms. All HMMs were Estimated using SAT.	11
Table 3: Number of Shared States and Mixtures for the ARL Dari Systems Created using Constrained Amounts of Training Data.	12
Table 4: Dari and Pashto WERs on the TRANSTAC Corpora.	17
Table 5: Mandarin CERs Obtain on the HUB4 Test Set, RT-03, and RT-04 using LMs Estimated on the HUB4 Train Set and Chinese Gigaword Corpus.	18
Table 6: Amount of Data used from the TDT4 Corpus (Used/total Hours); CER and SER Obtained by Comparing the Alignments and Recognizer Output.	20
Table 7: Comparison of WERs Obtained with the HDecode and Sphinx-4 Recognizer.....	22
Table 8: Example Langlink From Wikipedia.	37
Table 9: Example SWAT Database Records.	37
Table 10: SWAT Dabatase "City" Search Results.	37

THIS PAGE LEFT INTENTIONALLY BLANK

SUMMARY

This document provides a summary of work completed by SRA International under the work unit 71840871, Speech Interfaces for Multinational Collaboration, for the period 09 April 2009 to 30 September 2010 under contract FA8650-09-D-6939-0002.

Automatic Speech Recognition (ASR) systems were developed on the Army Research Laboratory (ARL) Dari corpus using the Hidden Markov Model (HMM) ToolKit (HTK). Several methods were investigated for improving the performance of this system, and these techniques were evaluated using varying amounts of training and test data. Speech data and transcripts from the Translation System for Tactical Use (TRANSTAC) Dari and Pashto corpora were segmented, formatted, verified, and partitioned into training and test partitions. These corpora were used to train ASR systems with HTK. Mandarin systems were developed with HTK on speech data that included only approximate transcripts. In addition, HTK was compared with the Rhine-Westphalian Technical University (RWTH) ASR system and the Sphinx-4 recognition engine.

Information Extraction and Retrieval components included video processing, English and Mandarin gender detection, metadata extraction using Lucene, Solr, and Tika, initial development of a searchable multimedia indexing system called Haystack, and exploiting multilingual relationships inherent in Wikipedia interlanguage links.

Finally, system administration support is necessary to maintain and upgrade the high performance computing environments supporting this research. A summary is provided for some of the more significant system administrations tasks completed under this work unit.

The authors would like to acknowledge the following groups: (1) Army Research Laboratory for the Dari speech corpus, (2) Microsoft and Cambridge University for their Hidden Markov Model (HMM) ToolKit (HTK), (3) Carnegie Mellon University (CMU) and Cambridge University for their Statistical Language Modeling (SLM) toolkit, (4) the International Computer Science Institute (ICSI) for their QuickNet software package, (5) Joe Frankel *et al.* for their Articulatory Feature (AF) classifiers, (6) Stanford Research Institute for their Language Modeling toolkit (SRILM), (7) the Royal Institute of Technology (KTH) for the Snack Sound toolkit, (8) the Rhine-Westphalian Technical University (RWTH) for their HMM software package, (9) the National Institute of Standards and Technology (NIST) for their Speech Recognition Scoring Toolkit (SCTK), (10) CMU, Sun Microsystems, Mitsubishi Electric Research Labs, Hewlett Packard, the University of California at Santa Cruz, and the Massachusetts Institute of Technology for the Sphinx-4 recognizer, (11) the Apache Software Foundation (ASF) for the Lucene search engine library, Solr search platform, Tika content analysis toolkit, PDFBox PDF library, and HTTP server, (12) the CentOS Project for the CentOS operating system, (13) the Oracle Corporation for the MySQL relational database management system, (14) the PHP Group for the PHP: Hypertext Preprocessor (PHP), (15) Larry Wall and the Perl Foundation for the Perl programming language, (16) the FFmpeg team for the FFmpeg software libraries and programs for handling multimedia data, (17) Systran for SYSTRAN-USG Enterprise Server Machine

Translation (MT) software, (18) the Wikimedia Foundation for the online Wikipedia encyclopedia available in over 272 languages and for making all the Wikipedia data available for download.

1.0 INTRODUCTION

This document provides a summary of work completed by SRA International under the work unit 71840871, Speech Interfaces for Multinational Collaboration, for the period 09 April 2009 to 30 September 2010 under contract FA8650-09-D-6939-0002.

Section 2 describes experiments and accomplishments in Automatic Speech Recognition and Information Extraction and Retrieval, as well as system administration tasks completed that support the research environment.

Section 3 summarizes conclusions drawn from the experiments and makes recommendations for future efforts.

2.0 EXPERIMENTS AND ACCOMPLISHMENTS

This section discusses experiments and accomplishments. Section 2.1 covers Automatic Speech Recognition, Section 2.2 covers Information Extraction and Retrieval, Section 2.3 covers SCREAM Wikipedia Aided Translation, and Section 2.4 covers System Administration support.

2.1 Automatic Speech Recognition

This section discusses the Automatic Speech Recognition (ASR) experiments performed. Section 2.1.1 describes the ASR systems created on the Army Research Laboratory (ARL) Dari corpus using the Hidden Markov Model (HMM) ToolKit (HTK). Several methods were investigated for improving the performance of this system, and these techniques were evaluated using varying amounts of training and test data. Section 2.1.2 describes how the Translation System for Tactical Use (TRANSTAC) Dari and Pashto corpora were formatted prior to developing ASR systems, and it presents the recognition results obtained for each language. Section 2.1.3 describes an ASR system developed for Mandarin Broadcast News and discusses how approximate transcripts were used to improve the performance of this system. Section 2.1.4 provides a comparison of HTK and the Rhine-Westphalian Technical University (RWTH) ASR system on a Mandarin Broadcast News recognition task. Section 2.1.5 compares the HDecode and Sphinx-4 recognition engines on six different ASR tasks and compares the Mel-Frequency Cepstral Coefficients (MFCCs) extracted by HTK and Sphinx-4. Finally, Section 2.1.6 summarizes the experiments performed, and Section 2.1.7 provides recommendations for future work.

2.1.1 ASR on the ARL Dari Corpus

This section discusses the ASR systems developed on the ARL Dari corpus. This corpus was collected by the ARL with support from the Air Force Research Laboratory (AFRL), and includes approximately 20 hours of read speech spoken by male speakers. Section 2.1.1.1 describes the baseline system and several methods that were investigated for improving the performance of this system. Next, Section 2.1.1.2 discusses how Multi-Layer Perceptrons (MLPs) trained on English speech data were incorporated into a Dari speech recognizer. Finally, Section 2.1.1.3 discusses the data constrained experiments performed.

2.1.1.1 ASR System Development

This section discusses the baseline ASR system and several methods that were investigated for improving the performance of this system—namely, using a multiple-state short-pause model, optimizing the number of shared states and mixture components, applying Speaker Adaptive Training (SAT), and estimating the models using a discriminative training criterion. The baseline acoustic models were state-clustered, across-word triphones trained using HTK [1].¹ The feature set consisted of 12 MFCCs, with Cepstral Mean Normalization (CMN), plus an energy feature. In addition, delta and acceleration features were included to form a 39 dimensional feature set. Decision tree clustering was performed using a clustering threshold of 500.0 and 40 questions that were derived from the phonetic descriptions included with the corpus. The number of

¹ Available at <http://htk.eng.cam.ac.uk>

mixture per state was chosen to minimize the Word Error Rate (WER) on the test set, and mixture incrementing was performed such that the number of mixture components used to model each HMM state was proportional to the number of training frames available for that state. The final system included 4706 shared states with an average of 14 mixture components per state.

Decoding was performed using the HTK large vocabulary continuous speech recognizer HDecode. A trigram Language Model (LM) was trained using the Carnegie Mellon University (CMU)-Cambridge Statistical Language Modeling (SLM) Toolkit [2].² The LM probabilities were estimated using only the training partition, but the vocabulary was expanded to include all words in the corpus. This system yielded a 26.5% WER on the test set.

Silence is typically modeled within an HTK system using both a three-state context-independent HMM and a single-state HMM. Figure 1 (A) shows the HMM prototype for the three-state model, and Figure 1 (B) shows the prototype for the single-state model. The three-state HMM, referred to as the *silence* model in this document, allows both the center state to be skipped and transitions between the final and initial emitting state. The single-state HMM, or *short-pause* model, shares its state distribution with the center state of the three-state model. This HMM allows its state to be skipped, thus producing no observations.

² Available at <http://www.speech.cs.cmu.edu/SLM/toolkit.html>

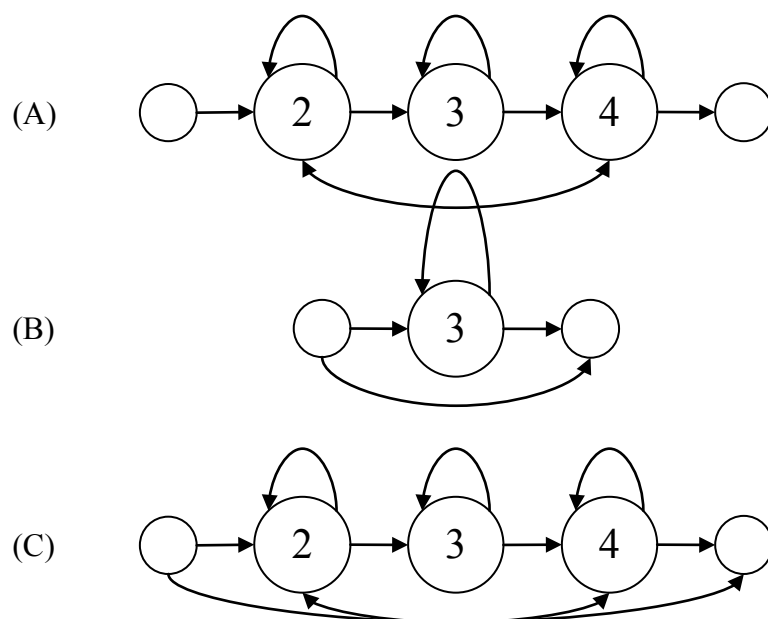


Figure 1: (A) Three-state Silence, (B) One-state Short-pause, and (C) Three-state Short-pause HMMs. Only States 2, 3, and 4 are Emitting, and the Arcs Show the Allowable Transitions

In an attempt to better model inter-word pauses and non-speech events, a second HMM system was trained using a three-state short-pause model. As in the case of the single-state short-pause HMM, all states can be skipped, and this HMM shares its state distributions with the three-state silence model. Figure 1 (C) shows the prototype for the three-state short-pause HMM. With the exception of the short-pause model prototype, the same procedure as described above was used to train and evaluate this system. The final set of HMMs included 4740 shared states with an average of 14 mixture components per state. This system yielded a 25.9% WER on the test partition with the HDecode recognizer. Compared to the HMM set that included a single-state short-pause model, this system yielded a 0.6% absolute improvement in WER. A three-state short-pause model was used for all remaining experiments discussed in this section.

In the experiments discussed above, the clustering threshold was guessed based on recognition results from other languages and corpora. To analyze the trade-off between the number of shared states and mixture components, 54 sets of acoustic models were developed on the Dari corpus. The HMMs were trained by varying the clustering threshold such that the number of shared states varied from 1000 to 5000, and the average number of mixture components per shared state varied from 12 to 32. Figure 2 shows the WERs obtained. Overall, we can see that using a lower number of shared states with more mixture components decreased the WER. For example, the system with 1500 shared states and an average of 28 mixture components yielded a 24.3% WER. Compared to the system trained without optimizing the number of shared states, this is a 1.6% absolute improvement in WER.

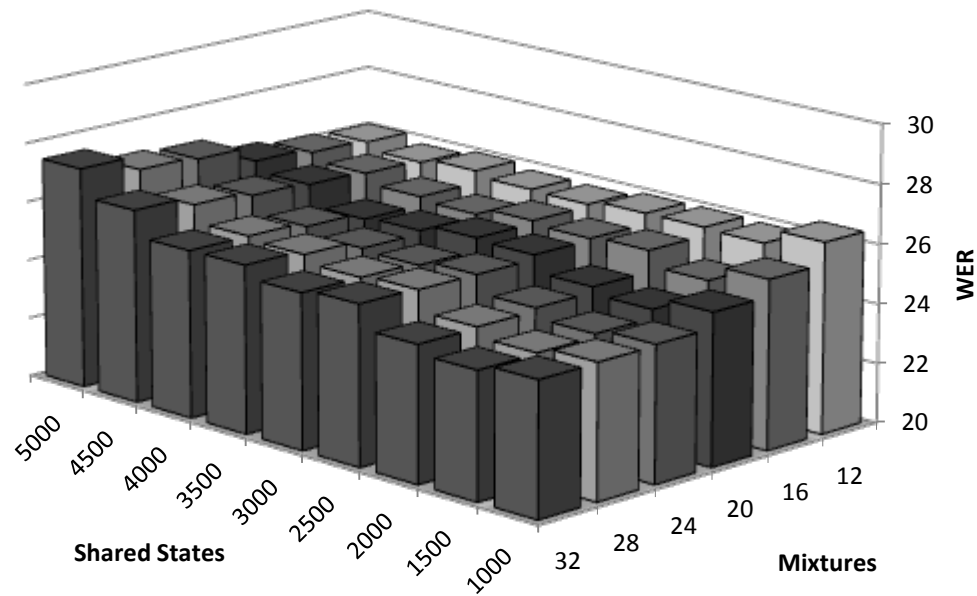


Figure 2: ARL Dari WERs using MFCCs with Varying Numbers of Shared States and Mixtures

Speaker Adaptive Training (SAT) [3] is a technique used to train speaker-independent acoustic models that integrates speaker normalization as part of the model estimation procedure. The procedure used for applying SAT and evaluating the test data can be summarized as follows:

1. Train multiple-mixture triphone models from the complete set of speakers
2. Compute a regression class tree
3. Estimate a set of transforms for each speaker
4. Re-estimate the models from Step 1 using the speaker transforms to adapt the features
5. Decode all test utterances using the initial models from Step 1
6. Estimate a set of transforms for each speaker using the SAT models from Step 4 and the hypothesized transcripts from Step 5
7. Re-decode each utterance using the SAT models from Step 4 and the speaker transforms from Step 6

The HMMs with 1500 shared states and an average of 28 mixture components were used for the initial models. Next, a regression class tree with 32 leaf nodes was computed. The regression class tree is used to group the mixture components into classes so that multiple transforms can be used for each speaker. Constrained Maximum Likelihood Linear Regression (CMLLR) [4] was used to compute a set of linear transforms for each speaker, and a state occupancy threshold of 1000.0 was used to determine the number of transforms estimated for each speaker. The final set of SAT HMMs were trained by applying two iterations of re-estimation with the transformed features. Two recognition passes were used to decode each utterance from the test set. First, the non-SAT models were used to generate initial transcripts for the test data. Next, these transcripts were used to estimate a set of transforms for each test speaker. Finally, the test utterances were

re-decoded using the SAT models and speaker transforms. Note that all data from a single speaker was used when computing the transforms; thus this method is not applicable to real-time processing. In addition, it was assumed that all speaker identities are known a priori. Applying SAT yielded a 21.7% WER, which is an improvement of 2.6% absolute compared to the non-SAT models.

The final method that was investigated for improving the performance of the baseline system was estimating the models using a discriminative training criterion. HMMs are normally trained using Maximum-Likelihood (ML) estimation. Whereas ML training only considers the correct transcript when adjusting the model parameters, discriminative training also considers competing hypotheses and attempts to adjust the model parameters such that the recognition errors are reduced. Thus in order to apply discriminative training, a set of competing hypotheses is needed for the training set. One method would be to consider all possible alternative word sequences; however, this is only feasible for very small vocabularies. The method employed by HTK is to recognize the training data and encode the best scoring hypotheses in a lattice. The lattices were computed using HDecode with a unigram LM created from the training set.

Discriminative training was applied using the Minimum Phone Error (MPE) criterion [5]. The HMMs with 1500 shared states and an average of 28 mixtures were used as the initial models. This system yielded a 22.9% WER, which is a 1.4% absolute improvement compared to the ML models. Discriminative training was also applied to the SAT models. This system yielded a 19.8% WER, which is a 1.9% absolute improvement compared to the SAT models estimated without discriminative training.

2.1.1.2 MLP Features

This section discusses how MLPs trained on English Articulatory Features (AFs) were incorporated into a Dari speech recognizer. The MLPs were trained by Frankel *et al.* on 2000 hours of conversational telephone speech [6].³ A total of eight MLPs were used to model the following AF groups: place, degree, nasality, lip rounding, glottal state, vowel, tongue height, and tongue frontness. The inputs consisted of Perceptual Linear Prediction (PLP) [7] coefficients, with energy, delta, and acceleration coefficients. All MLPs included a context-window of nine for the input; that is, each network used the feature vectors at times $t-4, t-3, \dots, t+3, t+4$ to classify the vector at time t . The number of hidden units for each MLP varied from 1200–2400, and the number of output units ranged from 3–23. The total number of output units from all MLPs was 64. The input features were computed using HTK, and all networks were evaluated using the International Computer Science Institute (ICSI) QuickNet software package.⁴

A feature vector was created from the MLP outputs using the following procedure. First, the MLPs were evaluated on the Dari corpus with their output activation functions removed. This was done so that the scores more closely approximated a Gaussian distribution. Next, a 64 dimensional feature vector was formed by concatenating the scores from the individual AF

³ A second set of MLPs were also trained in-house using the same procedure described in [6]. These networks yielded similar performance to the ones trained by Frankel *et al.*

⁴ Available at <http://www.icsi.berkeley.edu/Speech/qn.html>

detectors. Lastly, a Karhunen-Loève Transformation (KLT) was estimated on the Dari training partition, and the top 26 dimensions were appended to the MFCC feature set described in Section 2.1.1. This feature set is referred to as MFCC-MLP in the remainder of this report.

State-clustered across-word triphones were trained using HTK. Based on the results obtained in Section 2.1.1.1, short-pauses were modeled using a three-state HMM and a total of 54 acoustic models were trained to analyze the trade-off between the number of shared states and mixture components. Figure 3 shows the WERs obtained. Overall, we can see that this feature yielded a similar trend to the MFCC system; that is, using a lower number of shared states with more mixture components decreased the WER. For example, the system with 2000 shared states and an average of 28 mixture components yielded a 22.4% WER. It is interesting to note that whereas the MFCC system from Section 2.1.1.1 yielded the lowest WER with 1500 shared states, the MFCC-MLP feature set yielded the lowest WER with 2000 shared states.

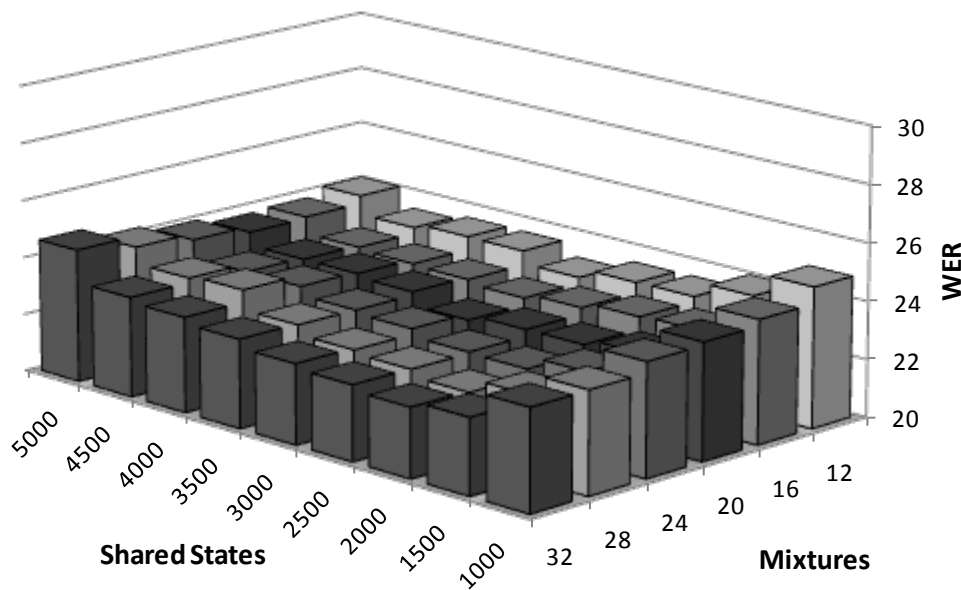


Figure 3: ARL Dari WERs using MFCC-MLP Features with Varying Numbers of Shared States and Mixture Components

SAT and discriminative training were applied using the same procedure as described in Section 2.1.1.1. The HMMs with 2000 shared states and an average of 28 mixtures per state were used as the initial models. SAT yielded a 20.9% WER, discriminative training yielded a 21.2% WER, and combining SAT with discriminative training yielded a 19.5% WER. Table 1 compares the results obtained in this section with those from Section 2.1.1.1. We can see from Table 1 that the MFCC-MLP features outperformed the MFCCs for all tasks, although the MFCC-MLP features provided the most benefit when SAT was not applied. For example, the MFCC-MLP features decreased the WER by 1.9% absolute when discriminative training and SAT were not applied; when discriminative training and SAT were applied, incorporating the MLP features reduced the WER by 0.3% absolute.

Table 1: Dari WER Comparison of MFCC and MFCC-MLP Features

SAT	MPE Training	MFCC	MFCC-MLP
no	no	24.3	22.4
yes	no	21.7	20.9
no	yes	22.9	21.2
yes	yes	19.8	19.5

2.1.1.3 Data Constrained Experiments

This section discusses the ASR experiments that were evaluated on the ARL Dari corpus using varying amounts of training and test data. When HMMs are estimated using SAT, transforms should be computed for each test speaker in order to yield an improvement in system performance [8]. In the experiments discussed in Sections 2.1.1.1 and 2.1.1.2, all speech data from each test speaker were used to compute the transforms. This section first analyzes how the WER varies based on the amount of speech data available for each test speaker.

Three different HMM systems were evaluated using varying amounts of test data: the MFCC system with SAT described in Section 2.1.1, the MFCC system with SAT and discriminative training from Section 2.1.1, and the MFCC-MLP system with SAT and discriminative training from Section 2.1.2. The first set of experiments evaluated all test files, but only used 6–120 seconds of speech data from each speaker to estimate the CMLLR transforms. For example, suppose test speaker *A* included 150 seconds of speech data and we wanted to evaluate the WER obtained when the maximum amount of speech data available for each speaker was 30 seconds. To accomplish this, the data from speaker *A* was partitioned into five subsets ($150 / 30 = 5$) and each subset was processed independently, as if the data from speaker *A* was actually spoken by five different speakers. The WER of each system is shown in Figure 4, where the systems that were estimated using a discriminative training criterion include the *MPE* label. The decrease in WER for the MFCC systems is less than 0.5% absolute when comparing systems that used 60 and 120 seconds of speech data. For the MFCC-MLP system, the decrease in WER is less than 0.5% absolute when comparing the systems that used 30 and 120 seconds of speech data.

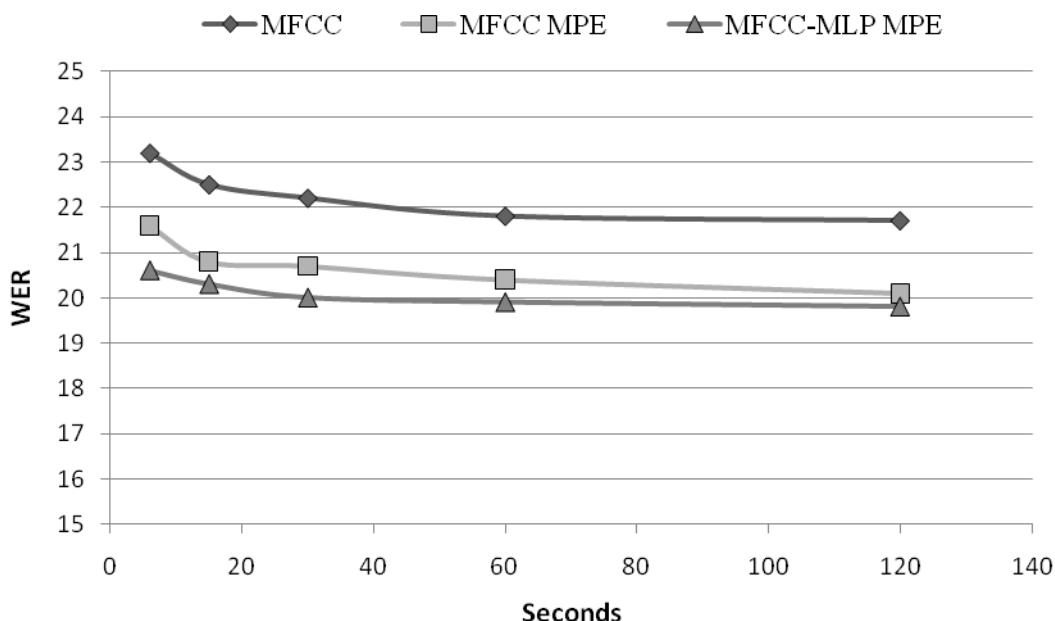


Figure 4: ARL Dari WERs Obtained using Varying Amounts of Test Data to Estimate the CMLLR Transforms. All HMMs were Estimated with SAT

A second experiment was performed that used an incremental approach to compute transforms. This method calculated a new set of transforms after each test utterance was processed; thus, the transforms were re-estimated using incrementally larger amounts of speech data. Note that this method could be used for real-time ASR since the lower limit on the amount of speech data required per-speaker is typically 5–10 seconds. The WER for each system is shown in Table 2. For comparison purposes, the results obtained using all available speech data are also included (referred to as *Batch* mode in Table 2). We can see from Table 2 that the incremental approach yielded slightly higher WERs than using all data to estimate a single set of transforms: the largest increase in WER was 0.4% absolute.

Table 2: ARL Dari WERs Obtained using an Incremental Approach for Computing the CMLLR Transforms. All HMMs were Estimated using SAT

Features	MPE Training	Batch	Incremental
MFCC	no	21.7	22.0
	yes	19.8	20.2
MFCC-MLP	yes	19.5	19.9

The remainder of this section investigates how the WER varies based on the amount of training data available. Training partitions were created from the ARL Dari corpus using 2.5, 5, 10, and 20 hours of speech data. HMM systems were created for each training partition using MFCC and MFCC-MLP features. As in previous sections, the number of shared states and mixture

components were chosen to minimize the WER on the test set.⁵ Table 3 shows the number of parameters chosen for each system. SAT and discriminative training were also applied to each system. In addition to the acoustic models, trigram LMs were created on each partition using the same procedure described in Section 2.1.1.1.

Table 3: Number of Shared States and Mixtures for the ARL Dari Systems Created using Constrained Amounts of Training Data

Training Hours	MFCC		MFCC-MLP	
	<i>Shared States</i>	<i>Mixtures</i>	<i>Shared States</i>	<i>Mixtures</i>
2.5	500	16	500	20
5.0	500	24	500	28
10.0	1000	28	1000	24
20.0	1500	28	2000	28

Figure 5 shows the WERs obtained with each system.⁶ The left-side of the chart shows the results obtained without SAT, and the right-side shows the results obtained with SAT. The systems that were estimated using a discriminative training criterion include the *MPE* label. As expected, increasing the amount of training data improved the performance of all systems. It is interesting to note that when SAT was not applied, the MFCC-MLP features outperformed the MFCC features for each task. When SAT was applied, however, the MFCC-MLP systems yielded worse performance than the MFCC MPE systems, and the MFCC-MLP MPE systems yielded little or no improvement over the MFCC MPE systems.

⁵ The number of shared states and mixtures was chosen using the HMMs estimated without SAT or discriminative training. Although not considered in this paper, note that the number of shared states and mixtures that yields the lowest WER may be different for models estimated with SAT or discriminative training.

⁶ This chart and the original interpretations of it were formed by Dr. Raymond Slyh of 711 HPW/RHXS.

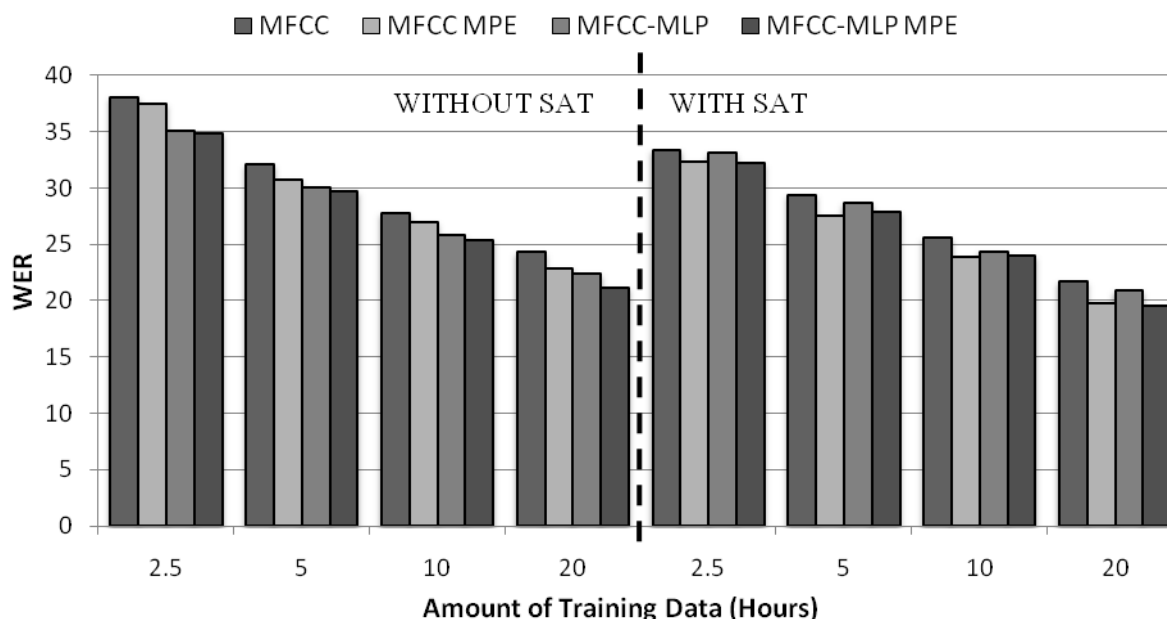


Figure 5: ARL Dari WERs using Varying Amounts of Acoustic and LM Training Data. The Left Side of the Chart Shows the WERs for each System without SAT, and the Right Side Shows the WERs for each System with SAT

A second set of experiments was conducted where the amount of acoustic model training data was constrained, but the LM was estimated from the 20 hour training partition. This scenario is of interest because whereas it is very time consuming and expensive to collect and transcribe acoustic speech data, LM training texts might be collected rather easily from sources such as the Internet. Figure 6 shows the WERs obtained with the baseline MFCC acoustic models and the MFCC acoustic models estimated using SAT and discriminative training.⁶ The label *20hr LM* is used to identify the systems that were evaluated with the LM estimated from the 20 hour train partition. Note that the MFCC and MFCC SAT MPE systems are repeated from Figure 5.

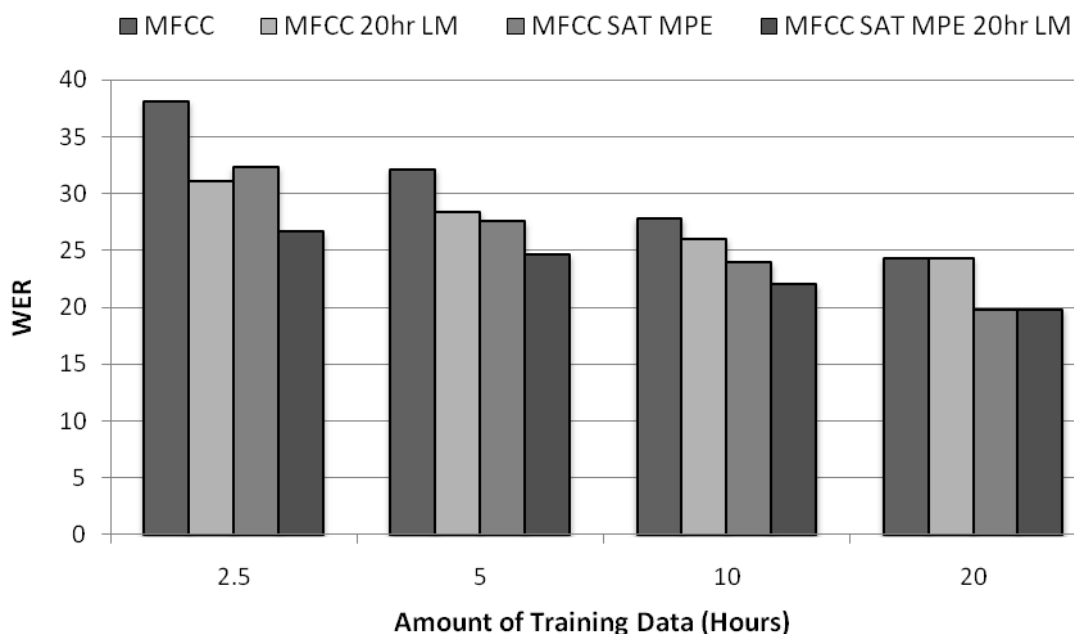


Figure 6: ARL Dari WERs using Varying Amounts of Training Data. Each HMM System was Evaluated using a Data Constrained LM, and an LM Estimated on 20 Hours of Data

Consider again the MFCC and MFCC SAT MPE systems that were developed using a constrained amount of acoustic and LM training data. From Figure 6 we can see that the 2.5 hour MFCC SAT MPE system yielded comparable performance to the 5 hour MFCC system. Similarly, the 5 hour MFCC SAT MPE system yielded comparable performance to the 10 hour MFCC system, and the 10 hour MFCC SAT MPE system yielded comparable performance to the 20 hour MFCC system. Thus by incorporating SAT and discriminative training we were able to reduce the amount of training data by 50% and still obtain comparable WERs.

As expected, using the 20 hour LM improved the performance of the systems trained on 2.5, 5, and 10 hours of data. From Figure 6 we can see that the MFCC SAT MPE 20hr LM system trained on 2.5 hours of acoustic data yielded comparable performance to the MFCC 20hr LM system trained on 10 hours of acoustic data. Similarly, the MFCC SAT MPE 20hr LM system trained on 5 hours of acoustic data yielded comparable performance to the MFCC 20hr LM system trained on 20 hours of acoustic data. Thus, when using the 20 hour LM, SAT, and discriminative training, we were able to reduce the amount of acoustic training data by 75% and still obtained comparable performance to the baseline systems.

2.1.2 Dari and Pashto ASR on the TRANSTAC Corpora

This section discusses how ASR systems were developed on the Dari and Pashto TRANSTAC corpora. The Dari corpus includes 158 hours of transcribed speech spoken by 210 speakers, and the Pashto corpus includes 183 hours of transcribed speech spoken by 181 speakers. There are two different collections of recordings for each database: the first consists of recordings between

an English interviewer, an interpreter, and a Dari or Pashto respondent; and the second collection consists of recordings between a Dari or Pashto interviewer and respondent. The TRANSTAC corpora were delivered without predefined training and test sets; thus, it was necessary to partition the data from each language into training and test sets prior to developing an ASR system.

A considerable amount of preprocessing was required before an ASR system could be developed for each language: the speech data had to be segmented, formatted, verified, and partitioned into the training and test sets. The following procedure was used to segment each conversation into utterance level audio and transcript files. First, all speech data from the English interviewers and all English speech segments spoken by the interpreters were ignored. This reduced the total amount of transcribed speech to 110 hours of Dari spoken by 192 speakers, and 130 hours of Pashto spoken by 156 speakers. Next, neighboring segments spoken by the same speaker were merged when possible. This was done in an effort to reduce the number of speech files and potentially improve the LM by increasing the frequency of higher order N-grams. Finally, punctuation marks and several symbols (used for speaker restarts, word fragments, mispronunciations, etc.) were removed from the texts.

To identify possible transcription errors, an ASR system was trained and evaluated on each language. The acoustic models were state-clustered across-word triphones, and the feature set consisted of 12 MFCCs with CMN plus energy, delta, and acceleration coefficients. Each set of HMMs included 5500 shared states with an average of 28 mixtures per state. Trigram LMs were created for each language using the Stanford Research Institute Language Modeling (SRILM) Toolkit [9], and decoding was performed using the HDecode recognizer. The Dari system yielded a 15.3% WER and the Pashto system yielded a 13.4% WER. Next, phoneme alignments were computed for each database and compared to the hypothesized phoneme sequences. Utterances with a Phoneme Error Rate (PER) greater than 30% were sequestered from each database. This reduced the total amount of transcribed speech to 99 hours of Dari and 129 hours of Pashto.

Training and test partitions were created for the Pashto corpus using the following procedure.⁷ First, a trigram LM was estimated for each of the 156 speakers, and all text data from each speaker was evaluated against every LM. This was done in an effort to find speakers with closely matched transcripts (*i.e.*, speakers who talked about similar topics). The normalized perplexity score for each speaker pair A and B was defined as follows

$$PP_{NORM}(A, B) = PP_A(W_B) - PP_A(W_A), \quad (1)$$

where $PP_A(W_B)$ is the perplexity obtained by evaluating all text from speaker B against the speaker A LM, and $PP_A(W_A)$ is the perplexity obtained by evaluating all text from speaker A against the speaker A LM. In an attempt to create more diverse test sets, all speakers with low scores were constrained to the training set. For a given threshold τ , each speaker A was constrained to the training set if they scored lower than τ against any other speaker B , that is

⁷ All text analysis was performed by Mr. Eric Hansen of 711 HPW/RHXS.

$$PP_{NORM}(A, B) < \tau \quad \text{or} \quad PP_{NORM}(B, A) < \tau \quad \text{for any} \quad B \neq A. \quad (2)$$

One partition was created with $\tau = 100$ and a second partition was created with $\tau = 70$. A third partition was also created with $\tau = 100$ and the additional constraint that condition 2 was true for more than one speaker B . The number of speakers constrained to each training set was as follows: 67 for the first partition, 12 for the second partition, and 34 for the third partition. Subject to these constraints, each partition was defined by randomly assigning 80% of the speakers to the training set and the remaining 20% to the test set.

Next, a Pashto ASR system was developed on each partition. The HMMs included 4000 shared states with an average of 20 mixtures per state, and decoding was performed with a trigram LM. WERs of 44.6%, 45.2%, and 44.9% were obtained on the first, second, and third partitions respectively. The largest difference in WER between the three partitions was 0.6%; thus it was concluded that for this corpus, the difference in transcripts between the speakers does not have a substantial effect on WER.

Based on the Pashto results, three partitions were created for the Dari corpus by randomly assigning 85% of the speakers to each training set and the remaining 15% to each test set. A Dari ASR system was developed on each partition. The HMMs included 5500 shared states with an average of 20 mixtures per state, and decoding was performed using a trigram LM. WERs of 46.8%, 48.0%, and 48.9% were obtained on the first, second, and third partitions respectively.

Updated versions of the transcripts were released in April and June of 2010. The same procedure as described above was used to process each corpus and sequester utterances with possible transcription errors. The updated database included 96 hours of transcribed Dari speech and 126 hours of transcribed Pashto speech. ASR systems were developed on the first Dari and Pashto partition using the revised transcripts. In addition, a fourth Pashto partition was created using a smaller pool of speakers for the test set to reduce the decoding time. Whereas the first three partitions used 20% of the speakers for the test set, the fourth partition used 10%. The following WERs were obtained: 45.7% on the first Dari partition (1.1% absolute improvement), 42.3% on the first Pashto partition (2.3% absolute improvement), and 41.9% on the fourth Pashto partition.

Finally, ASR systems that incorporated Heteroscedastic Linear Discriminant Analysis (HLDA) and discriminative training were developed on the first Dari partition and the fourth Pashto partition. The feature set consisted of 12 MFCCs, with CMN, plus energy, delta, acceleration, and third differential coefficients. An HLDA transform was estimated for each language using single-mixture monophone HMMs to reduce the feature dimension from 52 to 39. The Dari HMMs included 5500 shared states and the Pashto HMMs included 4500 shared states; both sets of models included an average of 20 mixtures per state. Discriminative training was applied using the same procedure described in Section 2.1.1.1.

The results for each language are shown in Table 4. HLDA yielded a 1.7% absolute improvement in WER on the Dari corpus, and a 1.8% absolute improvement in WER on the Pashto corpus. Applying discriminative training yielded an additional 3.0% and 3.2% absolute improvement in WER on the Dari and Pashto corpora respectively.

Table 4: Dari and Pashto WERs on the TRANSTAC Corpora

HLDA	MPE Training	Dari	Pashto
no	no	45.7	41.9
yes	no	44.0	40.1
yes	yes	41.0	36.9

2.1.3 Mandarin Broadcast News ASR

This section discusses the ASR systems created for Mandarin broadcast news. First, a baseline system was developed on the 1997 Mandarin Broadcast News (HUB4) [10] and Chinese Gigaword [11] corpora. HUB4 includes approximately 30 hours of transcribed speech from three different radio and television sources: Voice of America (VOA), China Central Television (CTV), and a commercial radio station (KAZN). The Chinese Gigaword corpus includes approximately 1.3 billion characters of newswire texts collected from three different sources: Central News Agency, Taiwan; Xinhua News Agency; and Zaobao Newspaper.

Next, this ASR system was used to generate time-aligned transcripts for the Topic Detection and Tracking (TDT4) Multilingual Broadcast News Corpus [12]. The TDT4 corpus includes approximately 200 hours of Mandarin audio with closed-captions, or approximate transcripts. These transcripts are a close match to what was actually spoken,⁸ but are only roughly time-aligned with the audio and do not include annotations for non-speech events (*e.g.*, speaker noise, music, background noise). The speech data were collected from five different radio and television sources: CTV, VOA, China National Radio (CNR), China Broadcasting System (CBS), and China Television System (CTS). A second set of HMMs was trained using HUB4 and TDT4.

The ASR systems discussed in this section were evaluated on the HUB4 test set, the 2003 National Institute of Standards and Technology (NIST) Rich Transcription task (RT-03) [13], and the 2004 NIST Rich Transcription task (RT-04) [14]. Each task consists of one hour of speech from radio and television sources: HUB4 includes 20 minutes from CTV, 27 minutes from VOA, and 13 minutes from KAZN; RT-03 includes one 12 minute excerpt each from CTV, VOA, CNR, CBS, and CTS; and RT-04 includes one 20 minute excerpt each from CTV, New Tang Dynasty Television (NTDTV), and Radio Free Asia (RFA). Section 2.1.3.1 describes the baseline ASR system developed on the HUB4 and Chinese Gigaword corpora. Section 2.1.3.2 discusses the procedure used to generate time-aligned transcripts for the TDT4 corpus, and describes the updated ASR system.

2.1.3.1 Baseline ASR System

This section describes the baseline ASR system that was developed on the HUB4 corpus. The acoustic models were state-clustered across-word triphones trained using HTK. The feature set

⁸ The TDT4 documentation states that ‘In general, the quality of these transcripts is quite good in terms of lexical accuracy’ (Available at http://www ldc.upenn.edu/Catalog/docs/LDC2005T16/content_summary.txt)

consisted of 12 MFCCs with CMN plus energy, pitch, delta, and acceleration coefficients. The pitch contour of the speech signal was computed using the Entopic Speech Processing System (ESPS) method implemented in the Snack ToolKit.⁹ Since continuous HMMs were used for all experiments described in this document, it was necessary to define pitch values over unvoiced segments. This was accomplished using the method described in [15].¹⁰ Discriminative training was applied using the same method described in Section 2.1.1.1. The final set of HMMs included 3000 shared states with an average of 16 mixtures per state.

The following procedure was used to format the Chinese Gigaword corpus. First, all text that was not enclosed in paragraph markers was removed. This was done in an effort to remove miscellaneous text data that did not correspond to a news story (*e.g.*, sports scores, stock prices, headlines, editorial notes). Next, the text was parsed into sentences by assigning sentence breaks at exclamation points, question marks, and periods. In an attempt to distinguish between periods used for end-of-sentence markers and acronyms, monetary amounts, web addresses, etc., a sentence break was only assigned at periods that were not preceded by English letters or digits. Since Chinese does not include explicit word boundaries, the character strings were parsed into words using the Linguistic Data Consortium (LDC) Chinese segmenter.¹¹ Finally, punctuation marks, several symbols, and sentences that included digits were removed. Note that sentences with digits were ignored because the pronunciation dictionary did not include digits.¹² The final text included 664 million words.

A trigram LM was estimated from the Chinese Gigaword corpus using the SRILM Toolkit. For comparison purposes, an LM was also estimated on the HUB4 training set. The HDecode recognizer was evaluated on the HUB4 test set, RT-03, and RT-04 using the discriminatively trained HMMs with each LM. Table 5 lists the Character Error Rates (CERs) obtained on each corpus, and Figure 7 presents the CERs obtained on each show. From Table 5, we can see that the Gigaword LM provided a substantial reduction in CER for each corpus: 2.9% absolute on the HUB4 test set, 12.4% absolute on RT-03, and 9.1% absolute on RT-04. From Figure 7, we can see that the Gigaword LM yielded lower CERs than the HUB4 LM for all shows; however, the reduction in CER was not uniform across shows. Furthermore, we can see that the CERs vary substantially from show-to-show. For example, the following CERs were obtained with the Gigaword LM: 9.3% on RT-03 CTV, 44.0% on RT-04 RFA, and 75.8% on RT-03 CTS.

Table 5: Mandarin CERs Obtain on the HUB4 Test Set, RT-03, and RT-04 using LMs Estimated on the HUB4 Train Set and Chinese Gigaword Corpus

LM	HUB4	RT-03	RT-04
HUB4	21.2	42.9	36.7
Gigaword	18.3	30.5	27.6

⁹ Available from the Royal Institute of Technology (KTH) at <http://www.speech.kth.se/snack>

¹⁰ This algorithm was implemented by Mr. Eric Hansen of 711 HPW/RHXS.

¹¹ Available at http://projects.ldc.upenn.edu/Chinese/LDC_ch.htm

¹² Although not considered in this paper, converting all digits to Chinese characters would have been a better alternative.

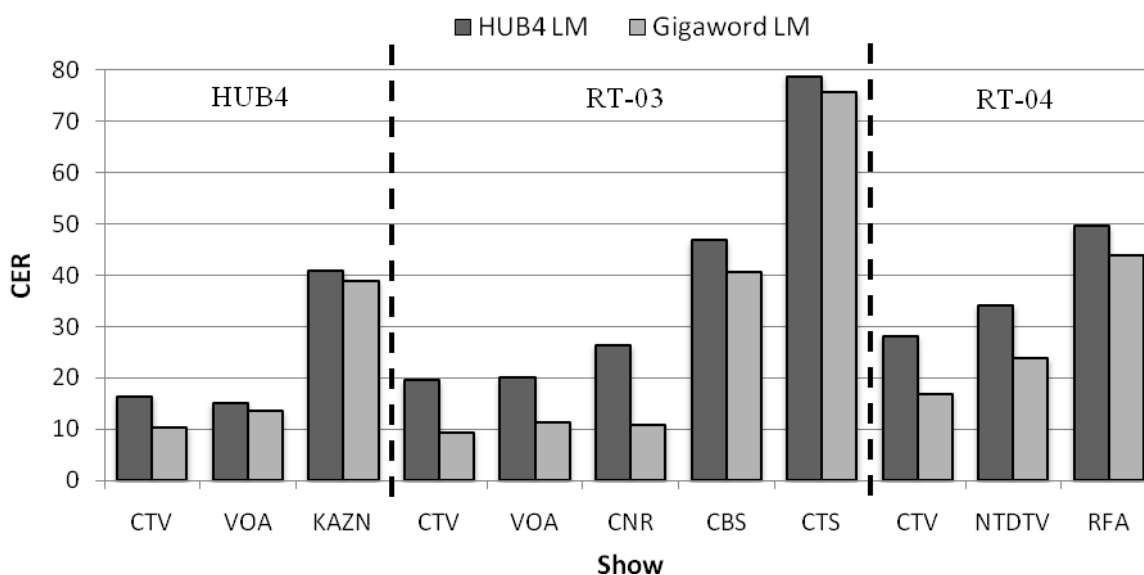


Figure 7: Mandarin CER on each Show from the HUB4, RT-03, and RT-04 Test Sets. The same HMMs were used for each Task with the HUB4 LM and Gigaword LM

Although all speech data used in these experiments were distributed with a 16 kHz sampling rate, the RT-03 CBS and CTS shows are band-limited to 4 kHz. In addition, these shows were recorded in Taiwan and may include dialectal differences. Note that if we ignore the CBS and CTS shows when computing the CER for RT-03, the ASR system with the Gigaword LM yielded a 10.4% CER. To determine the effect of the band-limiting, a second set of HMMs were trained on the HUB4 corpus with a 0–4 kHz pass-band. This system was evaluated on the RT-03 CBS and CTS shows using the Gigaword LM. A 36.7% CER was obtained on the CBS show (4.0% absolute improvement), and a 61.8% CER was obtained on the CTS show (14.0% absolute improvement).

2.1.3.2 ASR Training with Approximate Transcripts

This section describes how closed-caption filtering [16, 17] was used to create time-aligned transcripts for the TDT4 corpus. As mentioned in Section 2.1.3.1, the CBS and CTS shows were recorded in Taiwan and include band-limited speech. Due to these differences, only the CTV, VOA, and CNR shows were used from TDT4. After creating the transcripts, the HUB4 and TDT4 corpora were used to train a set of acoustic models.

The following procedure was used to generate time-aligned transcripts for the TDT4 corpus. First, the HMMs described in Section 2.1.3.1 were used to align the speech data. Each story was split into phrase level utterances by assigning a break point at any non-speech segment longer than 0.4 seconds. Next, an ASR system was evaluated on each utterance using the HMMs trained on HUB4, and a biased LM that was estimated from all closed-captions for that show. Finally, the alignments and the recognizer output were compared, and any utterances that did not match were sequestered from the database. Table 6 shows the final amount of usable data from each

show, along with the CER and Sentence Error Rate (SER) obtained by comparing the alignments and recognizer output. In all, a total of 75 hours was used from the TDT4 corpus.

Table 6: Amount of Data used from the TDT4 Corpus (Used/total Hours); CER and SER Obtained by Comparing the Alignments and Recognizer Output

Show	Hours	CER	SER
CTV	15/30	8.7	44.2
VOA	37/62	8.6	37.3
CNR	23/35	8.0	34.9

Acoustic models were trained on the on the HUB4 and TDT4 corpora using the same procedure described in Section 2.1.3.1. The final set of HMMs included 3000 shared states with an average of 20 mixtures per state. The HDecode recognizer was evaluated on the HUB4 test set, RT-03, and RT-04 using the Gigaword LM. Note that the transcripts from the TDT4 corpus were not pooled with the Gigaword texts because the TDT4 corpus is less than 1/1000th the size of the Gigaword corpus.¹³ As mentioned above, the CBS and CTS shows were recorded in Taiwan and include band-limited speech; thus, this system was not evaluated on these shows. Figure 8 shows the CERs obtained. For comparison purposes, the results obtained with the HUB4 acoustic models and Gigaword LM are also included.

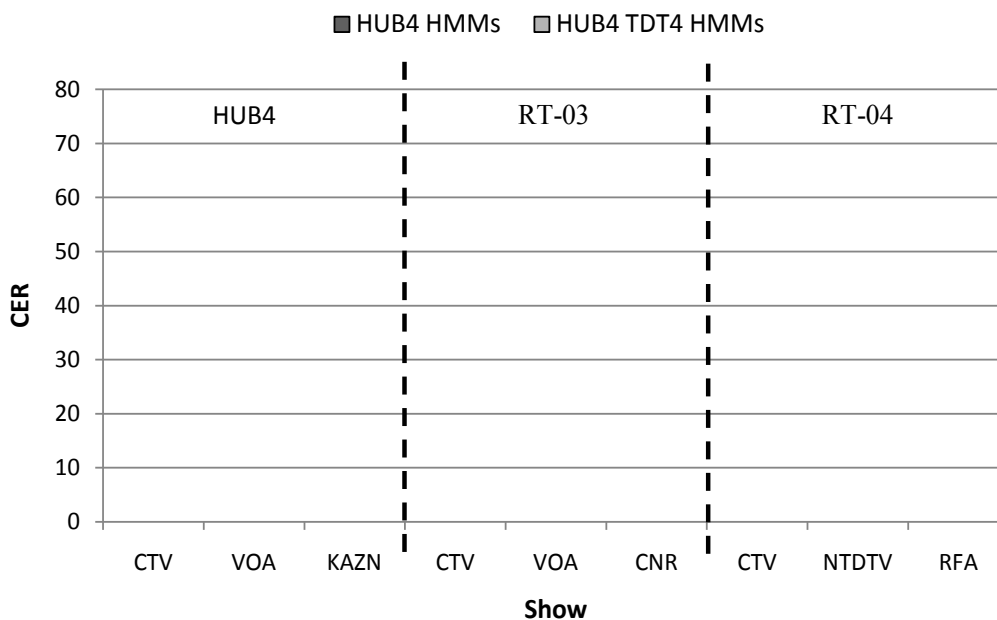


Figure 8: Mandarin CER on each Show from the HUB4, RT-03, and RT-04 Test Sets. The Gigaword LM was used for all Tasks, and the HMMs were Trained on the HUB4 Corpus, and the HUB4 TDT4 Corpus

¹³ Although not considered in this paper, a possible alternative would be to have trained an LM from the TDT4 texts and interpolated this LM with the Gigaword LM.

We can see from Figure 8 that incorporating the TDT4 training data yielded lower CERs for all shows. In terms of relative error, the largest improvements in system performance were 42% on RT-03 VOA, and 48% on RT-03 CNR. This is most likely attributed to the fact that both of these shows were also present in the TDT4 corpus. Furthermore, since broadcast news is largely dominated by a few anchors it is possible that some of the same speakers were present in both corpora. It is interesting to note that whereas the CTV shows were present in all corpora, the reduction in CER obtained by using the TDT4 data was less substantial for the CTV shows than for the VOA or CNR shows. Recall from Table 6 that the CER and SER were highest for the CTV shows: this implies that either the data was not as well matched, or the transcription quality of the CTV shows is worse. The overall CERs were as follows: 16.5% on the HUB4 test set (1.8% absolute improvement), 6.7% on RT-03 (3.7% absolute improvement), and 24.0% on RT-04 (3.6% absolute improvement).

2.1.4 RWTH ASR

The RWTH Aachen University ASR software package includes all of the necessary components for training and decoding HMMs [18].¹⁴ All experiments discussed here used version 0.4 of the toolkit, which includes many of the features available in the RWTH Aachen University internal version (most notable missing is support for discriminative training). This section summarizes several scripts that were developed for evaluating the software, and presents results obtained on the HUB4 Mandarin corpus.

The RWTH ASR software uses configuration files to define the system parameters, and Extensible Markup Language (XML) for database resources (*e.g.*, transcripts, pronunciation dictionaries, and decision tree questions). A Perl module was developed for manipulating the configuration files, and scripts were created to perform the following tasks: convert transcripts, dictionaries, and decision tree questions to XML format; evaluate the acoustic model trainer and recognizer across a grid of computers; and convert the output to a format readable by the NIST Speech Recognition Scoring Toolkit (SCTK).¹⁵

An HMM system was trained on the HUB4 corpus. The feature set consisted of 16 MFCCs with mean and variance normalization. As an alternative to differential coefficients, nine consecutive frames of MFCCs at times $t-4, t-3, \dots, t+3, t+4$ were concatenated to form the feature vector at time t , and Linear Discriminant Analysis (LDA) was used to reduce the feature dimension from 144 to 45. All HMMs included three states, except for silence, which was modeled using a single Gaussian. Global covariance matrices and transition probability matrices were used to model speech and non-speech sounds. Phonemes were modeled using within-word triphones, and state tying was performed using a Classification and Regression Tree (CART) [19]. The final set of HMMs included 4500 states with a total of 438,690 densities.

Decoding was performed on the HUB4 test set using the HUB4 and Gigaword LMs described in Section 2.1.3.1. A 21.2% CER was obtained with the HUB4 LM, and an 18.1% CER was obtained with the Gigaword LM. Recall from Section 2.1.3.1 that the HTK system yielded a

¹⁴ Available at <http://www-i6.informatik.rwth-aachen.de/rwth-asr>

¹⁵ Available at <http://www.itl.nist.gov/iad/mig/tools>

21.2% CER with the HUB4 LM, and an 18.3% CER with the Gigaword LM. Thus the RWTH ASR system yielded comparable performance to HTK.

2.1.5 Sphinx-4 Recognizer

Sphinx-4 is an HMM-based speech recognizer developed by CMU, Sun Microsystems, Mitsubishi Electric Research Labs, Hewlett Packard, the University of California at Santa Cruz, and the Massachusetts Institute of Technology (MIT) [20].¹⁶ Sphinx-4 is open-source software, and since it was written in Java, it is also platform-independent. This software was originally developed for use with acoustic models estimated using CMU Sphinx, although HTK models are also supported. Section 2.1.5.1 compares the HDecode and Sphinx-4 recognizers on six ASR tasks, and Section 2.1.5.2 investigates the differences between MFCC features computed using HTK and Sphinx-4.

2.1.5.1 HDecode and Sphinx-4

Although Sphinx-4 provides support for HTK models, initial experiments yielded very high error rates. In addition, functionality was not provided for loading tied-model lists or decoding with HMMs that included varying numbers of mixtures per state. For these reasons, a script was developed to convert HTK models to Sphinx-4 format, and the Sphinx-4 source code was modified to support HMMs with varying numbers of mixtures per state. In addition, a Java class was created for writing the hypothesized word sequences in HTK Master Label File (MLF) format.

To directly compare the HDecode and Sphinx-4 recognizers, identical ASR models were evaluated on English, Croatian, German, Dari, and Pashto. All HMMs were trained using HTK, and trigram LMs were estimated for each language using the SRILM Toolkit. The following tasks were evaluated: English read speech and conversational telephone speech, Croatian and German read speech, and Dari and Pashto conversational speech. The following corpora were used to train each system: Phase I and II of the Wall Street Journal (WSJ) corpus [21, 22] for English read speech, Fisher [23, 24] for English conversational telephone speech, GlobalPhone [25] for Croatian and German read speech, and TRANSTAC for Dari and Pashto conversational speech. The English conversational telephone speech system was evaluated on the Switchboard files from RT-03, and all of the other systems were evaluated on the test partition of their corpus. Table 7 shows the WERs obtained on each task; we can see that HDecode and Sphinx-4 yielded comparable performance.

Table 7: Comparison of WERs Obtained with the HDecode and Sphinx-4 Recognizer

	English (WSJ1)	English (RT-03)	Croatian	German	Dari	Pashto
HDecode	10.2	33.2	28.3	17.0	41.0	36.9
Sphinx-4	10.2	33.1	28.3	17.0	41.5	37.4

¹⁶ Available at <http://cmusphinx.sourceforge.net>

2.1.5.2 MFCC Feature Computation

In the experiments described in Section 2.1.5.1 all features were computed off-line using HTK. Since feature extraction is supported within Sphinx-4, HMMs were trained on GlobalPhone Croatian and German using HTK to estimate the model parameters and Sphinx-4 for feature extraction. This is useful for streaming audio and evaluating Sphinx-4 in real-time mode without any dependencies on HTK. The Sphinx-4 feature set included the same coefficients as the HTK features, namely, 13 MFCCs (including the zeroth coefficient), with CMN, plus delta and acceleration coefficients. All Croatian systems included 1500 shared states with an average of 20 mixtures per state, and the German systems used 3000 shared states with an average of 16 mixtures per state. The Sphinx-4 features yielded a 43.5% WER on Croatian and a 17.0% WER on German. Recall from Table 6 that the HTK features yielded a 28.3% WER on Croatian and a 17.0% WER on German. Although the same type of MFCCs were computed using HTK and Sphinx-4, there are differences in the way that the each software package calculates the coefficients. The remainder of this section investigates these differences.

The HTK MFCCs were calculated using the following procedure. First, the sampled speech waveform was blocked into 25 ms frames computed every 10 ms. The mean was removed from each frame, a pre-emphasis filter was applied, and a Hamming window was used to attenuate the discontinuities at the edge of each frame. Next, a filterbank was designed using 40 triangular filters spaced along the frequency axis to give approximately equal resolution on the Mel-scale [26]. The Fast Fourier Transform (FFT) was calculated for each frame, the magnitude of each FFT coefficient was multiplied by the corresponding filter gain, and the results for each filter were accumulated. The static MFCCs were calculated from the $\log_{10}(\cdot)$ of the filterbank sums using the Discrete Cosine Transform (DCT). If we denote the sum of each filter for the l^{th} frame as $m_l(j)$, then the MFCCs can be computed as follows

$$c_l(i) = \sqrt{\frac{2}{N}} \sum_{j=0}^{N-1} \log_{10} \{m_l(j)\} \cos\left(\frac{\pi}{N}(j-0.5)\right), \quad 0 \leq i \leq 12, \quad (3)$$

where $c_l(i)$ is the i^{th} static MFCC, and $N = 40$ is the number of filters in the filterbank. Next, the coefficients were filtered so that they had a similar range of values. Finally, CMN was applied and the differential coefficients were calculated using the following formula

$$dc_l(i) = \frac{\sum_{\theta=1}^{\Theta} \theta (c_{l+\theta}(i) - c_{l-\theta}(i))}{2 \sum_{\theta=1}^{\Theta} \theta^2}, \quad 0 \leq i \leq 12, \quad (4)$$

where $\Theta = 2$ for all experiments discussed in this paper. The delta coefficients were calculated directly from Equation (4), and the acceleration coefficients were calculated by substituting the delta coefficients $dc_l(i)$ for the static coefficients $c_l(i)$ in Equation (4).

Variance flooring is typically applied to an HMM system to prevent over-training. In HTK, the minimum value for any variance is set equal to the global variance of the training data times a variance scaling factor f . For HTK MFCCs, $f = 0.01$ is typically a reasonable choice. Upon inspection of the Croatian HMM definitions, however, it was found that 74% of variances were floored in the Sphinx-4 MFCC system, whereas less than 1% of the variances were floored in the HTK MFCC system. Consider the case where the l^{th} speech frame is all zeros. Since the magnitude of the FFT will be all zeros, the corresponding filterbank sums $m_l(j)$ will also be all zeros. Recall from Equation (3) that MFCCs are calculated from $\log_{10}\{m_l(j)\}$ and that $\log_{10}(0)$ is undefined. To handle this situation HTK floors all filterbank sums to 1.0, whereas Sphinx-4 approximates the $\log_{10}(0)$ as -100000. When computing MFCCs with Sphinx-4, a few frames of all zeros can substantially skew the mean and variance statistics. The Sphinx-4 feature extraction code was modified to floor the filterbank sums to 1.0, and an HMM system was re-trained on the GlobalPhone Croatian. This system yielded 28.4% WER.

In addition to handling frames of all zeros differently, it was also found that Sphinx-4 uses a different scaling factor when applying the DCT. Sphinx-4 computes MFCCs as follows

$$c_l(i) = \sum_{j=0}^{N-1} u(j) \log_{10}\{m_l(j)\} \cos\left(\frac{\pi i}{N}(j-0.5)\right), \quad u(j) = \begin{cases} \frac{1}{2N}, & j = 0 \\ \frac{1}{N}, & j \neq 0 \end{cases}, \quad 0 \leq i \leq 12. \quad (5)$$

Note that if we let $u(j) = \sqrt{2/N}$ for all j then Equations (5) and (3) are identical. A final difference between HTK and Sphinx-4 MFCCs is the method used for calculating differential coefficients. Sphinx computes delta and acceleration coefficients for the l^{th} frame as follows

$$dc_l(i) = c_{l+2}(i) - c_{l-2}(i) \quad (6)$$

$$d^2c_l(i) = [c_{l+3}(i) - c_{l-1}(i)] - [c_{l+1}(i) - c_{l-3}(i)], \quad (7)$$

where $0 \leq i \leq 12$. Note that if we expand Equation (4) for $\Theta = 2$, then the HTK delta and acceleration coefficients can be expressed in terms of the static MFCCs as follows

$$dc_l(i) = -0.2c_{l-2}(i) - 0.1c_{l-1}(i) + 0.1c_{l+1}(i) + 0.2c_{l+2}(i) \quad (8)$$

$$d^2c_l(i) = 0.04c_{l-4}(i) + 0.04c_{l-3}(i) + 0.01c_{l-2}(i) - 0.04c_{l-1}(i) - 0.1c_l(i) - 0.04c_{l+1}(i) + 0.01c_{l+2}(i) + 0.04c_{l+3}(i) + 0.04c_{l+4}(i), \quad (9)$$

Comparing Equations (6, 7) with Equations (8, 9) we can see that HTK uses a wider context window to calculate the differential coefficients. The Sphinx-4 source code was modified to compute the DCT using the same method as HTK, and Java classes were developed for applying liftering and calculating first-third order differential coefficients using the HTK method. The

following WERs were obtained using the updated Sphinx-4 MFCCs: 28.1% on Croatian (0.2% absolute improvement), and 17.1% on German (0.1% absolute increase in WER).

2.1.6 Summary

This section summarizes the ASR experiments performed. First, a baseline ASR system was developed on the ARL Dari corpus using HTK and the CMU-Cambridge SLM Toolkit. MFCC acoustic models were trained using ML estimation, and decoding was performed using a trigram LM. This system yielded a 26.5% WER. The following methods were investigated for improving the system performance: using a multiple-state short-pause model, optimizing the number of shared states and mixture components, applying SAT, and estimating the models using a discriminative training criterion. Applying all of these methods reduced the WER to 19.8%, which is a 6.7% absolute improvement. MFCC-MLP acoustic models were also trained using ML estimation, SAT, and discriminative training. Depending on the training strategy, this feature set yielded a 0.3–1.9% absolute improvement in WER over the MFCCs.

ASR systems were also developed on the ARL Dari corpus using varying amounts of training and test data. MFCC and MFCC-MLP systems were first evaluated using varying amounts of test data to compute the CMLLR transforms. It was found that using 120 seconds of speech per-speaker yielded little improvement over 60 seconds (less than 0.5% absolute improvement). An incremental approach was also investigated for applying CMLLR, whereby the transforms were re-estimated after each test utterance was processed. This method yielded slightly higher WERs than using all data to estimate a single set of transforms: the maximum increase in WER was 0.4% absolute.

Next, ASR systems were developed on the ARL Dari corpus using 2.5, 5, 10, and 20 hours of training data. MFCC and MFCC-MLP acoustic models were trained using ML estimation, SAT, and discriminative training. When SAT and discriminative training were not applied, the MFCC-MLP features outperformed the MFCCs for all tasks. When SAT and discriminative training were applied to the MFCC systems, however, the MFCC-MLP features yielded no benefit. It was also found that by applying SAT and discriminative training to the MFCC systems, it was possible to reduce the amount of training data by 50% and still obtain comparable performance to the baseline MFCC system (*i.e.*, HMMs trained using only ML estimation). Furthermore, by decoding with an LM estimated from the entire training set, it was possible to reduce the amount of acoustic model training data by 75% and obtain similar performance to the baseline MFCC system.

Dari and Pashto ASR systems were developed on the TRANSTAC corpora. First, the speech data from each corpus were segmented, formatted, verified, and partitioned into training and test sets. Text analysis was performed in an attempt to create more diverse test sets; however, it was found that the difference in transcripts between speakers did not have a substantial effect on WER. MFCC acoustic models were trained on each language using HTK, and decoding was performed using trigram LMs estimated with the SRILM Toolkit. HMMs trained using ML estimation yielded a 45.7% WER on Dari and a 41.9% WER on Pashto. Applying HLDA and discriminative training yielded a 41.0% WER on Dari (4.7% absolute improvement) and a 36.9% WER on Pashto (5.0% absolute improvement).

ASR systems were developed for Mandarin Broadcast News using HTK and the SRILM Toolkit. The baseline system was trained on the HUB4 and Chinese Gigaword corpora. The feature set included MFCCs, plus a pitch feature, and the HMMs were estimated using a discriminative training criterion. Decoding was performed using a trigram LM. The following CERs were obtained: 18.3% on the HUB4 test set, 30.5% on RT-03, and 27.6% on RT-04. It was discovered that the CBS and CTS shows from RT-03 were recorded in Taiwan and consisted of speech band-limited to 4 kHz. Ignoring the CBS and CTS shows when computing the CER for RT-03 yielded a 10.4% CER. A second set of HMMs was trained using a pass-band of 0–4 kHz. This system yielded a 36.7% CER on the RT-03 CBS show (4.0% absolute improvement) and a 61.8% CER on the RT-03 CTS show (14.0% absolute improvement).

Next, the full-bandwidth HMMs were used to generate time-aligned transcripts for the TDT4 corpus using closed-caption filtering. The CBS and CTS shows from TDT4 were ignored, and the resulting acoustic data were pooled with HUB4 and used to train an updated set of HMMs. This system yielded the following CERs: 16.5% on the HUB4 test set (1.8% absolute improvement), 6.7% on RT-03 (3.7% absolute improvement), and 24.0% on RT-04 (3.6% absolute improvement).

The RWTH Aachen University ASR software package was also used to train and evaluate HMMs on the Mandarin HUB4 corpus. A Perl module was developed for manipulating XML configuration files, and scripts were created to perform the following tasks: convert transcripts, dictionaries, and decision tree questions to XML format; evaluate the acoustic model trainer and recognizer across a grid of computers; and convert the output to a format readable by the NIST SCTL. Decoding was performed on the HUB4 test set using the LMs described above. This system yielded a 21.2% CER with the HUB4 LM, and an 18.1% CER with the Gigaword LM. Note that these results are comparable to HTK.

Finally, the Sphinx-4 recognizer was evaluated on six different ASR tasks. A script was developed for converting HTK models to Sphinx-4 format, and functionality was added to perform the following: allow HMMs with varying numbers of mixtures per state, compute MFCCs using the HTK method, and save the output in MLF format. The Sphinx-4 systems yielded similar performance to HDecode.

2.1.7 Recommendations for Future Work

On the ARL Dari corpus, it was found that MFCC-MLP features yielded no benefit over MFCCs when SAT and discriminative training were applied. It would be interesting to evaluate the MFCC-MLP features on several different languages with various recording conditions to see if similar results are obtained. The MLPs used in this work were trained on English AFs; thus, it may be worthwhile to investigate whether MLPs trained on Dari or MLPs trained on phone features yield similar results.

Approximately 14 hours of Dari and 4 hours of Pashto speech data were sequestered from the TRANSTAC corpora. This suggests that there may be transcription errors in the TRANSTAC corpora that need to be verified. In addition, other methods of data validation might prove to be useful. Since WERs of 41.0% and 36.9% were obtained on the Dari and Pashto test sets, improved modeling strategies should be investigated. Possible suggestions include Vocal Tract

Length Normalization (VTLN), SAT, collecting additional LM training texts from Internet blogs, and morphology-based language modeling.

A substantial improvement in CER was obtained on Mandarin using the Chinese Gigaword corpus. For simplicity, however, sentences with digits were ignored. Thus, it would be worthwhile to reprocess this corpus with a digit-to-character converter to use all available texts. An improvement in system performance was also obtained by applying closed-caption filtering to the TDT4 corpus for additional acoustic model training data. Alternative data selection methods, such as those based on confidence scores, could be investigated for generating the time-aligned transcripts. In addition, recall that the CBS and CTS shows were ignored from TDT4 when updating the acoustic models. Future work could consider developing a band-limited Mandarin ASR system for the Taiwan dialect. Finally, the TDT4 texts were not used to update the Gigaword LM because of the differences in corpora size. LM interpolation could be used for incorporating the TDT4 texts.

The RWTH ASR software package and the Sphinx-4 recognition engine could be useful for system combination (*e.g.*, using Recognizer Output Voting Error Reduction [27]) or semi-supervised learning. Incorporating functionality into Sphinx-4 for computing CMLLR transforms could yield substantial improvements in system performance.

2.2 Information Extraction and Retrieval

Haystack was created as a conglomerate tool for harnessing the SCREAM Laboratory's toolset to extract various metadata from multilingual media files and to allow for search and retrieval based on the extracted metadata. Figure 9 shows a concept diagram of potential capabilities for the Haystack system.

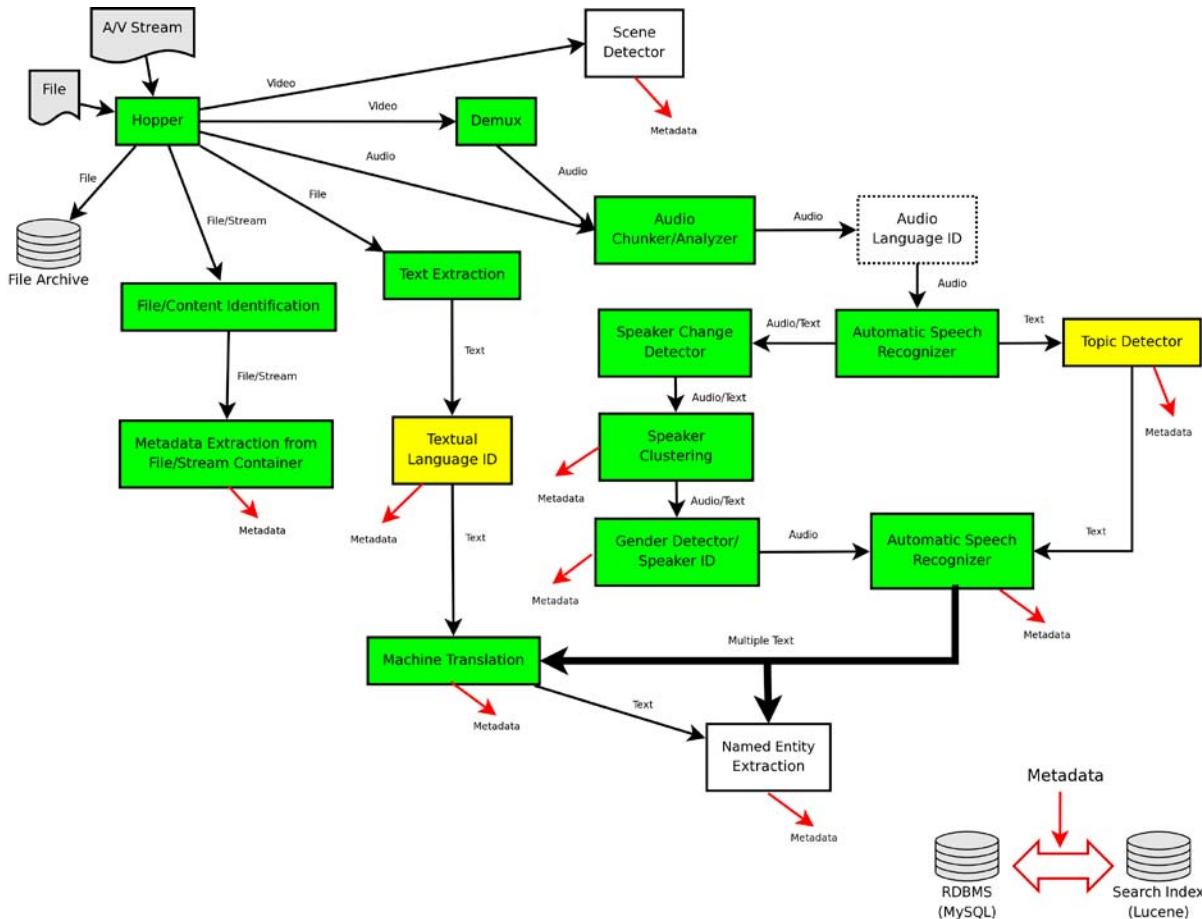


Figure 9: Haystack Concept Diagram

The goal under this task order was to develop an Initial Operating Capability (IOC) for the Haystack system. While many features are indicated in Figure 9, the IOC was only expected to provide document submission and search; automatic speech recognition (ASR) for English, Spanish, Arabic, and Mandarin; and Machine Translation (MT) from languages supported by Systran Webserver 5.0 to English. Additional features are expected to be added to the Haystack System as the result of future efforts.

2.2.1 Architecture

In order to store and search data generated by the Haystack system, a standard filesystem, a MySQL database, and a Lucene search index are used to store original data, derived data, and metadata.

A dedicated server running Linux, an Apache web server, a MySQL database server, Perl, PHP, Java, and Lucene/Solr were setup for development of the Haystack system. A basic diagram showing the Haystack architecture is shown in Figure 10.

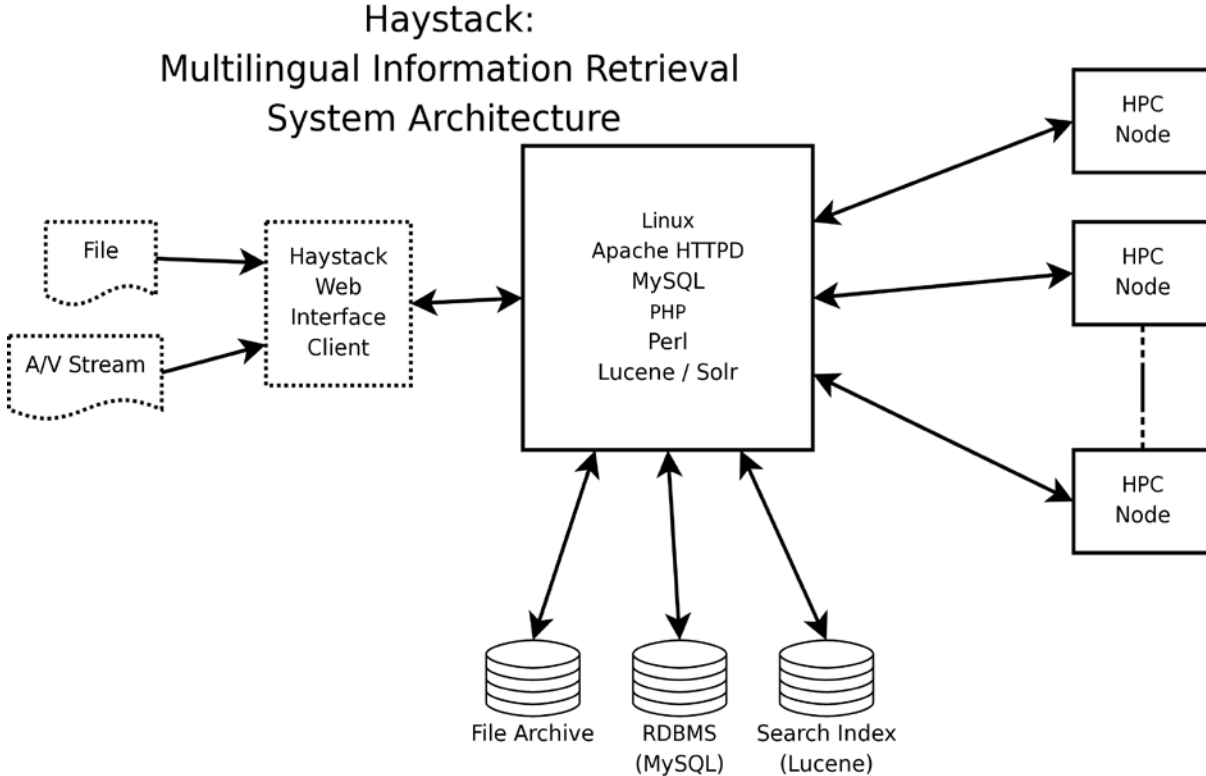


Figure 10: Haystack Architecture

The Haystack web-based user interface is provided by the Apache web server and PHP. Input data files and files created by the Haystack system are stored on the Linux file system with a directory structure that provides a unique directory for each document. Selected information about the documents processed by the Haystack system is stored in the MySQL database. The Apache Lucene search engine is used to create a searchable index of all the documents and derived data stored in the Haystack system. The Perl scripting language is used to control the back-end processing performed on each document. This processing is handled by additional GOTS, COTS, and SCREAM Lab developed software. Processing that is more computationally intensive may be performed on High Performance Computing (HPC) systems or networks.

2.2.2 User Interface

A web-based user interface was created that would enable a user to upload files, perform searches and browse files stored in the Haystack system. When a file is selected for upload, the server uses FFMPEG to detect parameters such as duration, audio/video bitrates, audio/video codecs, audio channels, and sample rates. These parameters are automatically populated in the user interface by an Asynchronous JavaScript and XML (AJAX) script. Users should enter the source, title, language, and keywords when submitting a document. Once the uploaded file is submitted, the web server stores selected information about the file to the MySQL database,

creates a unique document identifier, creates the unique file system folder for the document and saves the original document to the file system.

2.2.3 Back-End Processing

Once a document is submitted to the Haystack system through the web-interface, a Perl script is used to run the necessary processes to extract the available metadata from the document. Based on the file type, a sequential step of processes is performed on each document. Under this task order, the following processes were developed, assuming the languages were supported by current SCREAM Lab ASR and MT capabilities:

Video File: extract initial key frame image, extract audio, and perform ASR, MT, gender identification, segmentation, and Speaker Identification

Audio File: perform ASR, MT, gender identification, segmentation, and Speaker Identification

Text File: extract text from file and MT

The raw data from each process is stored in the document directory. When the back-end metadata processing is completed, a final process is run to add the new document and its metadata to the search index.

2.2.4 Lucene / Solr Search Index

Solr is able to emulate a SQL query on the database and create fields within the index that parallel the database fields and data by using a data-config.xml file. These fields are indexed and searchable. A 'text' field is also created as a catch-all so that a simple general query will use that instead of other, more specific fields. This will lead to a larger more generic return on query responses.

The data-config.xml calls on a JdbcDataSource type to connect to the Haystack MySQL database as shown in Figure 11. It creates a document with a generic query selecting all field columns stated within the document entity. The field columns are shown in Figure 12.

```
<dataSource type="JdbcDataSource" driver="com.mysql.jdbc.Driver"
url="jdbc:mysql://localhost/dbname" user="db_username" password="db_password"/>
```

Figure 11: Configuring the DataSource

```

<document>
  <entity name="document" query="select document_id, path, keywords, comments, lang_user, lang_detected, title, source,
created_timestamp, DATE_FORMAT(submitted_timestamp, '%Y-%m-%dT%H:%i:%sZ') AS submitted_timestamp, filename from document">
    <field column="document_id" name="id" />
    <field column="path" name="path" />
    <field column="keywords" name="hay_keywords" />
    <field column="comments" name="hay_comments" />
    <field column="lang_user" name="lang_user" />
    <field column="lang_detected" name="lang_detected" />
    <field column="title" name="hay_title" />
    <field column="source" name="hay_source" />
    <field column="created_timestamp" name="created_timestamp" />
    <field column="submitted_timestamp" name="submitted_timestamp" />
    <field column="filename" name="filename" />
  </entity>
</document>
</dataConfig>

```

Figure 12: Data-config.xml

Once the data-config file is created it can be tested without actually indexing an item, but can give feedback on record number returns. Once testing is complete, a data-config command can be sent to run the query and index the results.

Within the schema.xml, a catchall 'text' field was created that gathered these various Haystack fields and placed them within a text group for faster indexing and searching. Figure 13 provides a partial listing of the catchall 'text' schema. This allows for a quick query but not a detailed one. Searching for a specific term will bring results from all of the fields. The User Interface allows for field-specific searches as well as searches across all fields.

```

<copyField source="title" dest="text"/>
<copyField source="subject" dest="text"/>
<copyField source="description" dest="text"/>
<copyField source="comments" dest="text"/>
<copyField source="author" dest="text"/>
<copyField source="keywords" dest="text"/>
<copyField source="category" dest="text"/>
<copyField source="content_type" dest="text"/>
<copyField source="last_modified" dest="text"/>
<copyField source="links" dest="text"/>
<copyField source="path" dest="text"/>
<copyField source="db_id" dest="text"/>
<copyField source="source" dest="text"/>
<copyField source="lang_user" dest="text"/>
<copyField source="lang_detected" dest="text"/>

```

Figure 13: The Metadata is also Copied into a Catchall 'Text' Field

An update command also exists and is issued each step of the way throughout the Haystack metadata extraction process to keep information up-to-date for immediate queries. A nightly cron¹⁷ job is run for general updates and to catch any data that might have been missed.

¹⁷ Cron is a time-based job scheduler in Unix-like computer operating systems.

This section discusses the video processing software that was developed for Haystack. First, a program was created for converting between Y'CbCr and RGB color spaces. In the Y'CbCr color space, the Y' component stores information about the brightness of the image, and the CbCr components store the color information. RGB encodes the amount of red (R), green (G), and blue (B) in the image. This program expects Y'CbCr420p¹⁸ as input, which is a planar format that stores an entire frame of Y' pixels, followed by sampled frames of Cb and Cr pixels. Each Y value encodes a single pixel, whereas each Cb and Cr value encodes a 2×2 block of pixels. Figure 14 illustrates this for a single video frame. After converting each Cb and Cr frame to a full frame of pixels, the following equations were used to convert from Y'CbCr to RGB

$$\begin{aligned} R &= clip((298(Y-16)+409(V-128)+128) \gg 8) \\ G &= clip((298(Y-16)-100(U-128)-208(V-128)+128) \gg 8) \\ B &= clip((298(Y-16)+516(U-128)+128) \gg 8) \end{aligned} \quad (10)$$

Y0	Y1	Y2	Y3	Y4	Y5	Cb0 Cb1 Cb2 Cb3 Cb4 Cb5	Cr0 Cr1 Cr2 Cr3 Cr4 Cr5
Y6	Y7	Y8	Y9	Y10	Y11		
Y12	Y13	Y14	Y15	Y16	Y17		
Y18	Y19	Y20	Y21	Y22	Y23		

Figure 14: Y'CbCr420p Pixel Layout for each Frame. In this Example, the Resolution of the Video Frame would be 6 x 4 Pixels. Note that the Data Stream for this Video Frame would be {Y0, Y1, Y2, ..., Y23, Cb0, Cb1, Cb2, Cb3, Cb4, Cb5, Cr0, Cr1, Cr2, Cr3, Cr4, Cr5}

Next, a program was developed to convert an RGB video frame to Tagged Image File Format (TIFF). This is useful for displaying clips within the Haystack interface and applying Optical Character Recognition (OCR). The C library libTIFF¹⁹ was used to develop this program.

Finally, a program was developed for performing shot-boundary detection. Shot-boundary detection considers the problem of locating the times in a video where the camera changes to a new scene. The approach described in this paper uses histograms to calculate the difference between two consecutive frames, and then applies a threshold to classify the frame as either a

¹⁸ This term is often referred to as Y'UV420p, which describes the file format more accurately than the color space. See <http://en.wikipedia.org/wiki/YUV> for more information.

¹⁹ Available from <http://www.libtiff.org>

shot-boundary or non-shot-boundary [28]. Note that this method is most useful for detecting scene changes that are hard-cuts; that is, changes that are not a gradual transition across many frames. First, three histograms were created for each frame k and $k+1$ using the R, G, and B components of the video frame. Next, these histograms were compared by summing the Pearson's Chi-Square test statistic for each component

$$Z(k, k+1) = \sum_{i=R,G,B} \sum_{n=1}^N \frac{(H_{k+1}^i(n) - H_k^i(n))^2}{H_k^i(n)} \quad (11)$$

where N is the number of bins in each histogram, $H_k^i(n)$ is the number of $i = \{R, G, B\}$ pixels assigned to the n^{th} bin of frame k , and $H_{k+1}^i(n)$ is the number of $i = \{R, G, B\}$ pixels assigned to the n^{th} bin of frame $k+1$. If we hypothesize that frame k is a non-shot-boundary, then we can interpret $H_k^i(n)$ as the expected frequency and $H_{k+1}^i(n)$ as the observed frequency. Next, an adaptive threshold was used to classify the frames as either as shot-boundaries or non-shot-boundaries. The threshold was chosen based on the scores calculated from Equation (11). The k^{th} frame was classified as a shot-boundary if $Z(k, k+1)$ was the local maximum and was greater than α times larger than the second highest score that is

$$\begin{aligned} &\text{if } \left(\left[Z(k, k+1) = \max_{i=-M/2, \dots, M/2} \{Z(k+i, k+1+i)\} \right] \text{ and} \right. \\ &\left. \left[Z(k, k+1) > \alpha \max_{i=-M/2, \dots, -1, 1, \dots, M/2} \{Z(k+i, k+1, i)\} \right] \right) \text{ then } Z(k, k+1) \text{ is a shot boundary,} \end{aligned} \quad (12)$$

where M is the window length and α is a constant.

2.2.6 English and Mandarin Gender Detection

This section describes the gender-detectors that were developed for Haystack. Gaussian Mixture Models (GMMs) were trained on the 1996 English Broadcast News (HUB4) [29] and 1997 Mandarin Broadcast News corpora. The feature set consisted of 30 MFCCs, with Relative Spectra (RASTA) [30] processing, plus delta coefficients. After computing the features, the non-speech segments were removed using phoneme alignments generated with HTK. Next, a gender-independent 2048-mixture GMM was trained on each corpus using version 2.1 of the MIT-Lincoln Laboratory GMM software package [31]. Finally, male and female GMMs were created for each language by adapting the gender-independent models. A script was also developed for performing gender detection. This script evaluates the gender-dependent GMMs against each continuous speech segment in a file (specified via an MLF), and selects the best scoring model for each segment.

2.2.7 Summary

This section summarizes efforts in developing an Information Extraction and Retrieval system.

The Haystack project, a system to extract, translate, and present metadata from various multimedia files, was developed from conception to an IOC. The Haystack system uses a web-based front end to upload multimedia files. The files are analyzed and a number of operations are performed such as audio extraction, video frame extraction, gender detection, speaker identification, automatic speech recognition, and machine translation. The original file and all the resulting metadata are indexed and stored in the Haystack system. Within a few minutes of upload, the results are available and can be found in searches using the web-based user interface.

Software was developed to convert video frames between Y'CbCr and RGB color spaces, convert RGB video frames to TIFF, and perform shot-boundary detection. These programs were developed to support display of key frames, and detecting scene changes during videos.

Gender detection models were developed and trained for Mandarin and English using HUB4 Broadcast News corpora. To support the Haystack project, scripts were created to perform gender detection on an audio file using these models.

2.2.8 Recommendations for Future Work

Under this task order, the Haystack project has matured into a system capable of analyzing, storing, searching and displaying documents in Arabic, Chinese, English, Spanish, and Urdu. The analysis performed includes speaker identification, automatic speech recognition, and translation to English. While this progress represents a significant milestone for the Haystack system, more work remains to be completed under future efforts. Future work for the Haystack system could involve expansion in the capabilities of the user and search interface, expansion in the supported languages, enhancement of the architecture to allow processing of streaming multimedia, and the addition of processing components. Some potential processing components that have been identified are automatic language identification, named entity recognition, and topic detection.

2.3 SCREAM Wikipedia Aided Translation

Wikipedia is a free, web-based, collaborative, multilingual encyclopedia project supported by the non-profit Wikimedia Foundation.²⁰ As of August 19, 2010, Wikipedia contained 16,481,322 articles in 272 languages. The SCREAM Wikipedia Aided Translation (SWAT) effort was centered on the goal of exploiting connections between Wikipedia article titles in different languages [32]. Many Wikipedia pages contain interlanguage links. Interlanguage links are links from any page (most notably articles) in one Wikipedia language to one or more pages on the topic in another Wikipedia language.²¹ Figure 15 shows a graphical representation of the Wikipedia Interlanguage links.

²⁰ <http://en.wikipedia.org/wiki/Wikipedia>

²¹ http://en.wikipedia.org/wiki/Help:Interlanguage_links

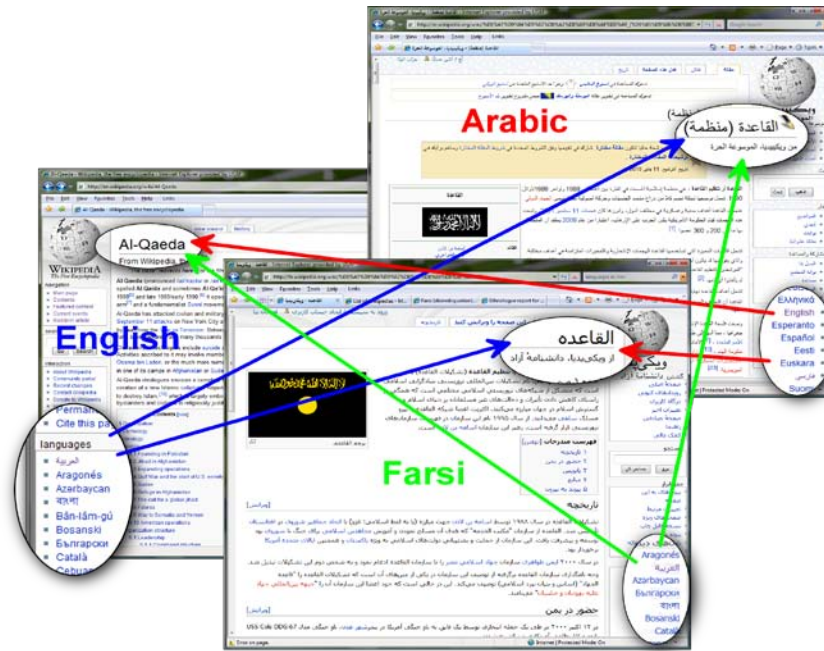


Figure 15: Wikipedia Interlanguage Links

A translation dictionary with over 117 million translation pairs was created by associating the title of each Wikipedia article in each language to the titles of articles in other languages by following the interlanguage links. To simplify the task of searching for and displaying entries in the SWAT database, a web-based user interface (UI) was developed.

2.3.1 Initial Approach

Due to the large size of Wikipedia and performance limitations, it was not feasible to extract the interlanguage link information by accessing the Wikipedia sites over the Internet. However, the Wikimedia Foundation offers free copies of all available content to interested users. Wikipedia data dumps were downloaded for each of 12 languages. Using the MediaWiki software and a web server, replicas of the 12 Wikipedias were setup on the local network. Although lacking images, these 12 Wikipedias were functioning copies of the original Wikipedia sites. However, the task of extracting useful relationships remained. The initial idea had involved extracting information from the Wikipedia web pages themselves, which would be an arduous task. With additional data dumps available from the Wikimedia Foundation, a more efficient technique was developed.

2.3.2 Refined Approach

A further examination of the Wikipedia data dumps revealed a number of useful MySQL database dumps. Specifically the *langlinks* and *page* tables for each language contained all the necessary information to build the SWAT database. Figure 16 shows how the Wikipedia data from the *langlinks* and *page* tables from Wikipedia are used to generate the *map* table for the SWAT database. The SWAT *map* table contains the resulting translation pairs and is created by associating the corresponding *page_title* (*entity1*) with the *ll_lang* (*lang2*) and *ll_title* (*entity2*)

from each *langlinks* record. The source language (*lang1*) is determined by the language of the *langlinks* table being processed.

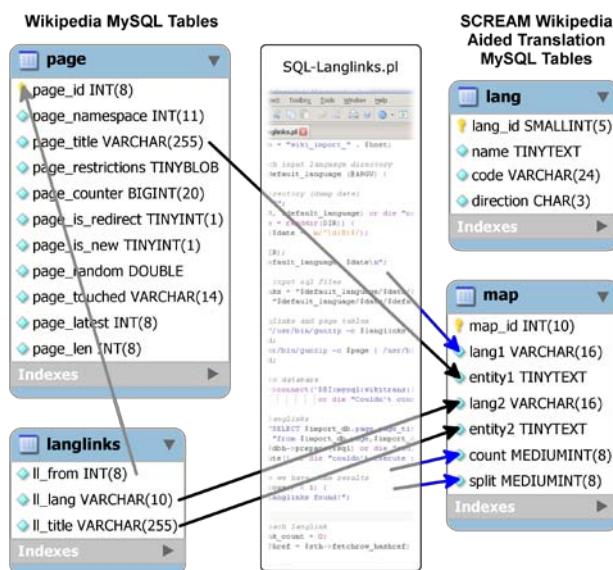


Figure 16: Wikipedia and SWAT Database Tables

A cursory examination of the Wikipedia titles indicated the possible utility of separating portions of titles delineated by parenthesis or a colon. If the parenthetical (or colon separated) information exists in both titles, additional translation entities could be obtained by creating additional associations between the parenthetical portions as well as the non-parenthetical portions.

A Perl script, *SQL-Langlinks.pl*, was developed to process each Wikipedia language and accumulate the results in the SWAT database. The Perl script loads the *page* and *langlinks* tables from a given language Wikipedia into a MySQL database and accomplishes the following steps for each record in the *langlinks* table:

- Check to see if the translation pair already exists in the *map* table. If the pair is already in the *map* table, *count* is incremented. If the pair is not already in the *map* table, it is inserted with a *count* of 1.
- If the *page_title* and *ll_title* both contain parenthesis, the parenthetical and non-parenthetical portions are separated, the corresponding parts are added to the database as additional pairs, and the *count* and *split* counters are incremented for each pair.
- If the *page_title* and *ll_title* both contain a colon, the titles are split at the colon, the corresponding parts are added to the database as additional links, and the *count* and *split* counters are incremented for each pair. If either *page_title* or *ll_title* has more than one colon, the second and any additional colons are ignored.

An example langlink from the English Wikipedia is shown in Table 8. The resulting entries in the map table of the SWAT database are shown in Table 9.

Table 8: Example Langlink from Wikipedia

page_title	ll_from	ll_title
Luxembourg (city)	Es	Luxemburgo (ciudad)

Table 9: Example SWAT Database Records

map_id	lang1	entity1	lang2	entity2	Count	split
1957026	en	Luxembourg (city)	es	Luxemburgo (ciudad)	1	0
1748157	en	City	es	ciudad	1	1
1858073	en	Luxembourg	es	Luxemburgo	1	1

In this example, we can see that separate treatment of the parenthetical and non-parenthetical portions of the title created useful and accurate information. However that is not always the case. Expanding on this analysis, the Table 10 represents a search of the SWAT database for Spanish translations of the English word “city”.

Table 10: SWAT Database "City" Search Results

map_id	lang1	entity1	lang2	entity2	count	split
1692769	en	city	es	Kazajistán	2	2
1706600	en	city	es	municipio	2	2
1721965	en	City	es	Israel	2	2
1748157	en	city	es	ciudad	37	35
1818775	en	City	es	Columbia Británica	4	4
1958643	en	City	es	Ecuador	2	2
1959853	en	City	es	Portugal	2	2
2049619	en	City	es	Nueva York	2	2
2074607	en	City	es	Turquía	2	2

Some inaccurate results are caused by associating the parenthetical information in the English and Spanish titles. An example of this is illustrated by the English Wikipedia page for “Langley, British Columbia (city)”²² which has a langlink to the Spanish Wikipedia page, “Ciudad de Langley (Columbia Británica)”. An accurate association is illustrated by the English Wikipedia

²² [http://en.wikipedia.org/wiki/Langley,_British_Columbia_\(city\)](http://en.wikipedia.org/wiki/Langley,_British_Columbia_(city))

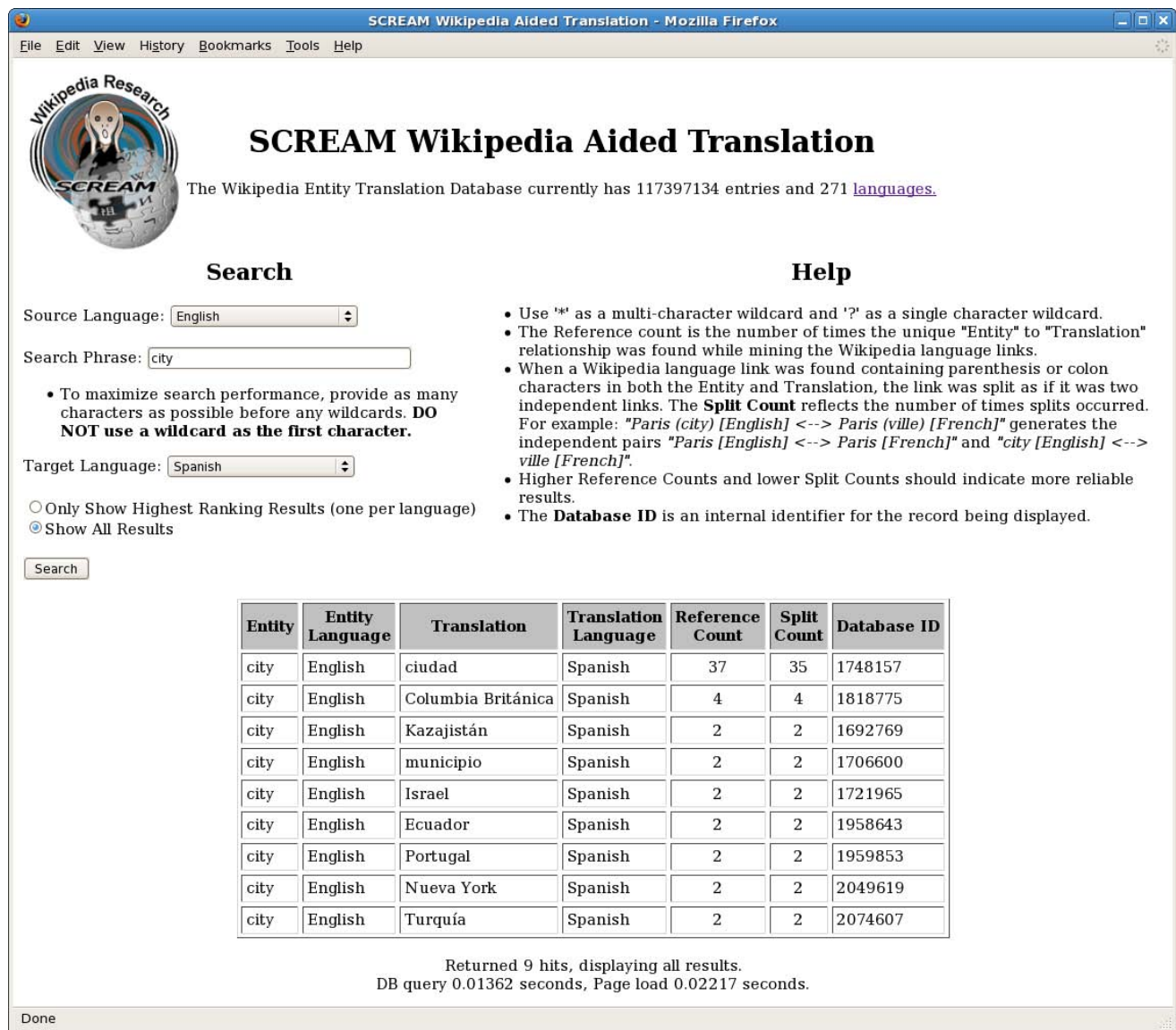
page for “Cabinda (city)”²³ which has a langlink to the Spanish Wikipedia page, “Cabinda (ciudad)”.

The *count* and *split* parameters can be used to provide a confidence about the search results. The correct result in Table 10 has a count of 37. This means the association between “city” and “ciudad” occurred 37 times when processing the Wikipedia Interlanguage links. Because the *count* parameter is two more than the *split* parameter, we know there were two original Wikipedia titles that associated “city” and “ciudad”. In this case the English Wikipedia page, “City” and the Spanish Wikipedia page, “Ciudad” are the original titles. It follows that 35 accurate associations of “city” and “ciudad” occurred from parenthetical (or colon separated) information. Adding up the *split* parameters for the inaccurate results, only 14 inaccurate associations were generated from parenthetical (or colon separated) information. Using this example, it would be reasonable to conclude that the separation of the parenthetical (or colon separated) information was beneficial. Due to the wide variety of data in Wikipedia this conclusion may or may not be reasonable in general. It would be reasonable to assign a higher confidence to results with a higher *count* parameter. In addition the confidence should be higher if the *count* parameter is higher than the *split* parameter as this indicates associations from original Wikipedia titles.

2.3.3 Web Interface

To simplify searching and displaying entries in the SWAT database, a web-based UI was developed using the Apache HTTP webserver, and the PHP Hypertext Preprocessor (PHP). This web-based user interface is illustrated in Figure 17.

²³ [http://en.wikipedia.org/wiki/Cabinda_\(city\)](http://en.wikipedia.org/wiki/Cabinda_(city))



SCREAM Wikipedia Aided Translation

The Wikipedia Entity Translation Database currently has 117397134 entries and 271 [languages](#).

Search

Source Language:

Search Phrase:

- To maximize search performance, provide as many characters as possible before any wildcards. **DO NOT use a wildcard as the first character.**

Target Language:

☐ Only Show Highest Ranking Results (one per language)
☒ Show All Results

Help

- Use '*' as a multi-character wildcard and '?' as a single character wildcard.
- The Reference count is the number of times the unique "Entity" to "Translation" relationship was found while mining the Wikipedia language links.
- When a Wikipedia language link was found containing parenthesis or colon characters in both the Entity and Translation, the link was split as if it was two independent links. The **Split Count** reflects the number of times splits occurred. For example: "Paris (city) [English] <--> Paris (ville) [French]" generates the independent pairs "Paris [English] <--> Paris [French]" and "city [English] <--> ville [French]".
- Higher Reference Counts and lower Split Counts should indicate more reliable results.
- The **Database ID** is an internal identifier for the record being displayed.

Entity	Entity Language	Translation	Translation Language	Reference Count	Split Count	Database ID
city	English	ciudad	Spanish	37	35	1748157
city	English	Columbia Británica	Spanish	4	4	1818775
city	English	Kazajistán	Spanish	2	2	1692769
city	English	municipio	Spanish	2	2	1706600
city	English	Israel	Spanish	2	2	1721965
city	English	Ecuador	Spanish	2	2	1958643
city	English	Portugal	Spanish	2	2	1959853
city	English	Nueva York	Spanish	2	2	2049619
city	English	Turquía	Spanish	2	2	2074607

Returned 9 hits, displaying all results.
DB query 0.01362 seconds, Page load 0.02217 seconds.

Figure 17: SWAT Web-Based User Interface

The SWAT UI translates search input into appropriate SQL queries, submits the queries to the MySQL database and displays the results. The user selects source language, translation language, keywords and ranking options for each search. For the source and translation languages, a wildcard can be selected to indicate all possible languages. Wildcards can also be used for the search phrase. A '*' character is used as a multi-character wildcard and the '?' character is used as a single character wildcard. Caution should be used when placing wildcards near the beginning of the search phrase as this minimizes the effectiveness of the database indexes, causing slow database queries. Since the database contains over 115 million rows, the database indexes greatly increase search performance even if only a few characters precede the wildcard(s).

The "languages" link in the second header line of the SWAT UI can be used to display statistics on the Wikipedia database dumps included in the SWAT database. The displayed information

includes the date the information was dumped from Wikipedia and the number of translation pairs generated for each language.

2.3.4 Creating (or Recreating) a SWAT Database

To build a SWAT database, the following resources are recommended:

- High performance MySQL server with 16GB of RAM and 50GB or more available hard disk space. A Linux operating system is preferred.
- Perl and Perl DBI

To use the SWAT UI, the following resources are necessary:

- Web Server, Apache is preferred
- PHP
- MySQL database server

The following instructions are intended to be used to create (or recreate) a SWAT database.

2.3.4.1 Create The MySQL Database

Several files containing SQL commands are used to create the tables in the SWAT database. The table defining the available languages is populated from a text file when the *lang* table is created. This text file needs to be copied into the database directory beforehand. The commands below illustrate the creation of a SWAT database, a database used for importing Wikipedia dumps, and a mysql account used to access the databases during processing.

```
mysqladmin create swat
mysqladmin create wiki_import
cp wiki-lang-list.csv /var/lib/mysql/swat
mysql swat < lang.sql
mysql swat < map.sql
mysql < create-wiki-user.sql
```

2.3.4.2 Download Wikipedia Dumps

The Wikipedia dumps can be downloaded from <http://download.wikimedia.org>. The dumps are organized by language and date. The directory for each language is created by the language code concatenated with the string “wiki”. The language directory contains a subdirectory for each dump. These subdirectories are named with 8 digit numbers representing the dump date in YYYYMMDD format. There is an additional subdirectory named “latest”. To build the SWAT database the “page” and “langlinks” dumps are needed. Using the English Wikipedia and a September 4th, 2010 dump date as an example, the download URLs are:

<http://download.wikimedia.org/enwiki/20100904/enwiki-20100904-langlinks.sql.gz>
<http://download.wikimedia.org/enwiki/20100904/enwiki-20100904-page.sql.gz>

While the “latest” directories may be useful for some purposes, the date of the actual Wikipedia dump is lost. It is preferred to download the Wikipedia dumps from the dated URLs as shown above. The Perl script used to build the SWAT database saves these dates in the database.

Download the “page” and “langlinks” dumps for all desired languages. Save the files in a directory structure created with the language codes then date codes. The following abbreviated list of files demonstrates the directory structure:

```
aa/20100310/aawiki-20100310-langlinks.sql.gz
aa/20100310/aawiki-20100310-page.sql.gz
...
zu/20100309/zuwiki-20100309-langlinks.sql.gz
zu/20100309/zuwiki-20100309-page.sql.gz
```

A Python based Wikipedia database dump download tool is available at <http://github.com/babilen/wp-download/>, however this tool has recently stopped working. The wp-download script fails while doing comparisons of the dump subdirectories in an attempt to find the latest dump. This is probably due to the recent addition of the “latest” directories in the dump subdirectories. Since “latest” is not numeric, Python crashes when trying to compare it to the subdirectories named with the dump date.

2.3.4.3 Building the SWAT Database

Building the SWAT database is a very computationally intensive process. Wikipedia dumps from March 2010 generated over 117 million unique records with a 10GB map table. Processing the Wikipedia dumps to build the SWAT database should be done on 64-bit MySQL server with at least 16GB of memory. The database server should be configured for very large innodb tables and query caches disabled. If the MySQL database server cannot keep the map table in memory, the processing will be significantly slower. On a dual quad core IBM System x3550 M2 system with 32GB of RAM, building a full (all languages) SWAT database took 216 hours. Two helpful utilities for adjusting database parameters to improve performance are MySQLTuner²⁴ and MySQL Performance Tuning Primer Script²⁵.

To build the SWAT database, the SQL-Langlinks.pl Perl script is executed for each language. The script should be executed from the top of the directory structure described in the previous section. For example:

```
SQL-Langlinks.pl en | /bin/gzip -c > ./en.log.gz
```

This command processes the interlanguage links for the English Wikipedia and captures the script output (stdout) into a gzip compressed log file. For progress indication, the script will also display numbers to the screen (stderr) as interlanguage links are processed. The log file will contain all the records added to the database. In order to build the complete SWAT database, this command must be repeated for all 271 languages. There is a Bourne shell script available, process-all.sh, which will process all 271 languages.

²⁴ <http://mysqltuner.com/>

²⁵ <http://www.day32.com/MySQL/>

2.3.4.4 *Generating Additional Indices and Language Counts*

The SWAT database will already have indices on the *entity1* and *entity2* columns. When building the database, these indices are necessary since the database must be continuously queried for existing translation pairs. In order to improve the performance for searches on particular languages or high counts, additional indices are recommended. The following MySQL command will create the additional indices.

```
Mysql swat  
Alter table map add index(lang1(3)), index(lang2(3)), add index(count);
```

Due to the size of the SWAT database, creating these indices may take a considerable amount of time.

The language statistics page of the SWAT UI displays the number of translation pairs that exist for each language. After the SWAT database is built, run the following command to generate the translation pair counts for each language:

```
SQL-Lang-Counts.pl
```

2.3.5 **Summary**

The SCREAM Wikipedia Aided Translation (SWAT) project successfully demonstrated the use of multilingual Wikipedia data to generate an enormous word/phrase translation utility for 271 languages. Harvesting the multilingual relationships between the linked titles between Wikipedias of different languages generated a database with over 117 million translation pairs. Using PHP and MySQL, a web-based user interface was developed to search the SWAT database. Instructions were provided to create or re-create a SWAT database. The scripts developed to generate the initial SWAT database and database dumps from the desired Wikipedias are required.

2.3.6 **Recommendations for Future Work**

With the amount of data available from Wikipedia, this research is just one example from a myriad of possibilities. One of the challenges with Wikipedia data is separating the valuable data from the superfluous fragments. The *count* and *split* parameters in the SWAT database provide a simple yet effective method of scoring results that helps identify the more valuable data. Further analysis and more sophisticated processing might improve the quality of the results.

Additionally, the Wikipedias contain extensive category lists that are often adequately populated. In a manner similar to the SWAT database creation, these Wikipedia categories could be exploited to create a Named Entity Recognition (NER) tool that uses literal data. While unlikely to outperform linguistic grammar based or statistical model based NER systems, it could supplement these more advanced NER systems while requiring little work by experienced computational linguists or large amounts of manually annotated training data.

2.4 System Administration Support

System administration support was provided to maintain the computational efficacy of the SCREAM Laboratory in order to support Speech Processing and Recognition (SPaRe) research.

The significant system administration tasks accomplished under this task order are listed below. Some less significant system administration tasks, such as routine system maintenance and user support, also accomplished under this task order may not be listed.

- Installed, configured, and tested write performance of a new 18TB SATA RAID array vs. existing Ultra-320 RAID arrays.
- Updated puppet software and scripts.
- Supported major network and computer reconfiguration during and after remodeling of the SCREAM Laboratory.
- Setup and configured 10 new Dell R610 computer systems.
- Investigated software to record video from computer based Free-To-Air (FTA) satellite cards. Developed scripts to select channels and stream live video from FTA satellite cards.
- Edited 107 hours of Croatian video that was recorded by the Virage system to remove non-Croatian segments from the beginning and/or end of the video segments, and eliminate corrupted video.
- Investigated use of older ATI Radeon 8500 DV video cards to capture analog Cable Television (CATV) signals.
- Setup and configured FreeSwitch VOIP software.
- Setup a Linux, Apache, MySQL, Perl, and PHP (LAMP) web server for the Haystack project.
- Evaluated future backup strategies and technologies to replace failing SDLT tape library.
- Upgraded CentOS Linux systems as new releases became available.
- Setup new Amanda backup server resulting in faster and more reliable backups.

3.0 CONCLUSIONS AND RECOMMENDATIONS

This document summarized work completed by SRA International, Inc. during the period April 2009 to September 2010.

Automatic Speech Recognition (ASR) systems were created with the Army Research Laboratory (ARL) Dari corpus and the Translation System for Tactical Use (TRANSTAC) Dari and Pashto corpora. An ASR system was also developed for Mandarin broadcast news. Finally, performance comparisons were made between several ASR systems. These accomplishments are summarized in greater detail in section 2.1.6

With the ARL Dari corpus, it was found that MFCC-MLP features yielded no benefit over MFCCs when SAT and discriminative training were applied. It would be interesting to evaluate the MFCC-MLP features on several different languages with various recording conditions to see if similar results are obtained. The MLPs used in this work were trained on English AFs; thus, it may be worthwhile to investigate whether MLPs trained on Dari or MLPs trained on phone features yield similar results.

ASR systems were trained and evaluated on the un-partitioned TRANSTAC corpora, and all utterances with a PER greater than 30% were sequestered from the database. PERs greater than 30% were obtained on approximately 14 hours of Dari and 4 hours of Pashto speech data. This suggests that there may be transcription errors in the TRANSTAC corpora that need to be verified. In addition, other methods of data validation might prove to be useful. Since WERs of 41.0% and 36.9% were obtained on the Dari and Pashto test sets, improved modeling strategies should be investigated. Possible suggestions include Vocal Tract Length Normalization (VTLN), SAT, collecting additional LM training texts from the Internet, and morphology-based language modeling.

A substantial improvement in CER was obtained on Mandarin using the Chinese Gigaword corpus. For simplicity, however, sentences with digits were ignored. Thus, it would be worthwhile to reprocess this corpus with a digit-to-character converter to use all available texts. An improvement in system performance was also obtained by applying closed-caption filtering to the TDT4 corpus for additional acoustic model training data. Alternative data selection methods, such as those based on confidence scores, could be investigated for generating the time-aligned transcripts. In addition, recall that the CBS and CTS shows were ignored from TDT4 when updating the acoustic models. Future work could consider developing a band-limited Mandarin ASR system for the Taiwan dialect. Finally, the TDT4 texts were not used to update the Gigaword LM because of the differences in corpora size. LM interpolation could be used for incorporating the TDT4 texts.

The RWTH ASR software package and the Sphinx-4 recognition engine could be useful for system combination (*e.g.*, using Recognizer Output Voting Error Reduction [27]) or semi-supervised learning. Incorporating functionality into Sphinx-4 for computing CMLLR transforms could yield substantial improvements in system performance.

Information Extraction and Retrieval components were developed to support video scene change detection, extracting information from various documents, and creating web-based searchable indexes from document content and metadata. A system, called Haystack, to extract, translate, and present metadata from various multimedia files, was developed from conception to an Initial Operational Capability (IOC). These developments are summarized in greater detail in section 2.2.7.

Under this task order, the Haystack project has matured into a system capable analyzing, storing, searching and displaying documents in Arabic, Chinese, English, Spanish, and Urdu. The analysis performed includes speaker identification, automatic speech recognition and translation to English. While this progress represents a significant milestone for the Haystack system, more work remains to be completed under future efforts. Future work for the Haystack system could involve expansion in the capabilities of the user and search interface, expansion in the supported languages, enhancement of the architecture to allow processing of streaming multimedia, and the addition of processing components. Some potential processing components that have been identified are automatic language identification, named entity recognition, and topic detection.

Software was developed to build a large word/phrase translation utility for 271 languages using multilingual data from Wikipedia. These developments are summarized in greater detail in section 2.3.5.

With the amount of data available from Wikipedia, this research is just one example from a myriad of possibilities. One of the challenges with Wikipedia data is separating the valuable data from the superfluous fragments. The *count* and *split* parameters in the SWAT database provide a simple yet effective method of scoring results that helps identify the more valuable data. Further analysis and more sophisticated processing might improve the quality of the results.

Additionally, the Wikipedias contain extensive category lists that are often adequately populated. In a manner similar to the SWAT database creation, these Wikipedia categories could be exploited to create a Named Entity Recognition (NER) tool that uses literal data. While unlikely to outperform linguistic grammar based or statistical model based NER systems, it could supplement these more advanced NER systems while requiring little work by experienced computational linguists or large amounts of manually annotated training data.

REFERENCES

1. Cambridge University Engineering Department, "The HTK Book," 2009 (Available at <http://htk.eng.cam.ac.uk>).
2. P. Clarkson and R. Rosenfeld, "Statistical Language Modeling using the CMU-Cambridge Toolkit," in *Proceedings of Eurospeech*, Rhodes, Greece, September, 1997.
3. T. Anastasakos *et al.*, "A Compact Model for Speaker-Adaptive Training," in *Proceedings of the International Conference in Spoken Language Processing*, Philadelphia, Pennsylvania, October 1996.
4. V. Digalakis *et al.*, "Speaker Adaptation Using Constrained Estimation of Gaussian Mixtures," *IEEE Transactions on Speech and Gaussian Mixtures*, Vol. 3, pp. 357–366, 1995.
5. D. Povey and P. Woodland, "Minimum Phone Error and I-Smoothing for Improved Discriminative Training," in *Proceedings of the International Conference of Acoustics, Speech, and Signal Processing*, Orlando, Florida, 2002.
6. J. Frankel *et al.*, "Articulatory Feature Classifiers Trained on 2000 hours of Telephone Speech," in *Proceedings of Interspeech*, Antwerp, Belgium, August 2007.
7. H. Hermansky, "Perceptual Linear Predictive (PLP) Analysis of Speech," *Journal of the Acoustical Society of America*, Vol. 87, No. 4, pp. 1738–1752, 1990.
8. T. Anastasakos *et al.*, "Speaker Adaptive Training: A Maximum Likelihood Approach to Speaker Normalization," in *Proceedings of the International Conference of Acoustics, Speech, and Signal Processing*, Munich, Germany, 2007.
9. A. Stolcke, "SRILM – An Extensible Language Modeling Toolkit," in *Proceedings of the International Conference on Spoken Language Processing*, Denver, Colorado, 2002.
10. S. Huang *et al.*, "1997 Mandarin Broadcast News Speech (HUB4-NE)," *Linguistic Data Consortium*, Philadelphia, 1998 (Available at <http://www ldc.upenn.edu>).
11. D. Graff *et al.*, "Chinese Gigaword Second Edition," *Linguistic Data Consortium*, Philadelphia, 2005 (Available at <http://www ldc.upenn.edu>).
12. S. Strassel *et al.*, "TDT4 Multilingual Broadcast News," *Linguistic Data Consortium*, Philadelphia, 2005 (Available at <http://www ldc.upenn.edu>).
13. J. Fiscus *et al.*, "2003 NIST Rich Transcription Evaluation Data," *Linguistic Data Consortium*, Philadelphia, 2007 (Available at <http://www ldc.upenn.edu>).
14. LDC2005E73 and LDC2005E74
15. C. J. Chen, *et al.*, "New Methods in Continuous Mandarin Speech Recognition," in *Proceedings of Eurospeech*, Rhodes, Greece, 1997.
16. L. Lamel *et al.*, "Lightly Supervised and Unsupervised Acoustic Model Training," *Computer Speech and Language*, Vol. 16, 2002.
17. J. Ma *et al.*, "Unsupervised Training on Large Amounts of Broadcast News Data," in *Proceedings of the International Conference of Acoustics, Speech, and Signal Processing*, Toulouse, France, 2006.
18. D. Rybach *et al.*, "The RWTH Aachen University Open Source Speech Recognition System," in *Proceedings of Interspeech*, Brighton, U.K., September, 2009.
19. K. Beulen *et al.*, "State-tying for context-dependent phoneme models," in *Proceedings of the European Conference on Speech Communications*, Rhodes, Greece, 1997.

20. W. Walker *et al.*, “Sphinx-4: A Flexible Open Source Framework for Speech Recognition,” SMLI TR2004-0811, Sun Microsystems Inc., 2004.
21. J. Garofolo *et al.*, “CSR-I (WSJ0) Complete,” *Linguistic Data Consortium*, Philadelphia, 2007 (Available at <http://www ldc upenn edu>).
22. “CSR-II (WSJ1) Complete,” *Linguistic Data Consortium*, Philadelphia, 1994 (Available at <http://www ldc upenn edu>).
23. C. Cieri *et al.*, “Fisher English Training Part I,” *Linguistic Data Consortium*, Philadelphia, 2004 (Available at <http://www ldc upenn edu>).
24. C. Cieri *et al.*, “Fisher English Training Part II,” *Linguistic Data Consortium*, Philadelphia, 2005 (Available at <http://www ldc upenn edu>).
25. T. Schultz, “GlobalPhone: a Multilingual Speech and Text Database Developed at Karlsruhe University,” in *Proceedings of the International Conference on Spoken Language Processing*, Denver, Colorado, 2002.
26. S. Stevens and J. Volkmann, “The Relation of Pitch to Frequency: A Revised Scale,” *American Journal of Psychology*, Vol. 153, pp. 329–353, 1940.
27. J. Fiscus, “A Post-Processing System to Yield Reduced Word Error Rates: Recognizer Output Voting Error Reduction (ROVER),” in *Proceedings of the IEEE Workshop on Automatic Speech Recognition and Understanding*, Santa Barbara, California, 1997.
28. A. Hanjalic, “Shot-Boundary Detection: Unraveled and Resolved?,” *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 12, No.2, 2002.
29. D. Graff *et al.*, “1996 English Broadcast News Speech (HUB4),” *Linguistic Data Consortium*, Philadelphia, 1997 (Available at <http://www ldc upenn edu>).
30. H. Hermansky and N. Morgan, “RASTA Processing of Speech,” *IEEE Transactions on Speech and Audio Processing*, Vol. 2. No. 4, pp. 578–589, 1994.
31. D. Reynolds *et al.*, “Speaker Verification using Adapted Gaussian Mixture Models,” *Digital Signal Processing*, Vol. 10, pp. 19–41, 2000.
32. A. Richman and P. Schone, “Mining Wiki Resources for Multilingual Named Entity Recognition,” *Proceedings of Association for Computational Linguistics: Human Language Technologies*, Columbus, OH, June 2008.

LIST OF ACRONYMS & GLOSSARY

AF	articulatory feature
AFRL/RHXS	Air Force Research Laboratory/Human Effectiveness Directorate, Anticipate & Influence Behavior Division, Sense-making & Organizational Effectiveness Branch
AJAX	Asynchronous JavaScript and XML, a technique used in web application development
Amanda	Amanda, previously known as Advanced Maryland Automatic Network Disk Archiver is an open source computer archiving tool that is able to back up data residing on multiple computers on a network.
ARL	Army Research Laboratory
Apache	The Apache Software Foundation is a group that supports open source software development projects.
API	Application Programming Interface
Apple	Apple Inc. is an American multinational corporation that designs and markets consumer electronics, computer software, and personal computers.
ASR	automatic speech recognition
AT&T	AT&T is a telecommunications firm that was created from the merger of Southwestern Bell Corporation & American Telephone and Telegraph Corp.
CART	classification and regression tree
CATV	Cable Television
CBS	China Broadcasting System
CentOS	CentOS is a community-supported, mainly free software operating system based on Red Hat Enterprise Linux.
CER	character error rate
CMLLR	constrained maximum likelihood linear regression
CMN	cepstral mean normalization
CMU	Carnegie Mellon University
CNR	China National Radio
COTS	Commercial Off-The-Shelf
CTS	China Television System
CTV	China Central Television
CURL	cURL is a computer software project providing a library and command-line tool for transferring data using various protocols
DCT	discrete cosine transform
Dell	Dell Inc is a American multinational information technology corporation that develops, sells and supports computers and related products and services.
DLIPS	Digital Library Input Processing System
DoD	Department of Defense
FFMPEG	FFmpeg is a free software / open source project that produces libraries and programs for handling multimedia data.
FFT	fast Fourier transform

Fisher	an English corpus of conversational telephone speech
FreeSWITCH	FreeSWITCH is a free/open source software communications platform for the creation of voice and chat driven products.
FTA	Free-to-air (FTA) describes television (TV) and radio services broadcast in clear (unencrypted) form, allowing any person with the appropriate receiving equipment to receive the signal and view or listen to the content without requiring a subscription.
ESPS	Entropic Speech Processing System
GlobalPhone	a multilingual text and speech database
GMM	Gaussian mixture model
GOTS	Government Off-The-Shelf
HTML	HyperText Markup Language
HPC	High-performance computing (HPC) uses supercomputers and computer clusters to solve advanced computation problems.
HPW	Human Performance Wing
IBM	International Business Machines (IBM) is an American multinational computer, technology and IT consulting corporation.
ICER	Information Operations Cyber Exploitation Research
ICSI	International Computer Science Institute
IOC	Initial Operational Capability (IOC) is the state achieved when a capability is available in its minimum usefully deployable form.
IR	information retrieval
IWSLT	International Workshop on Spoken Language Translation
Haystack	An internal SCREAM Lab project to integrate the various SCREAM Lab capabilities into a system to index, analyze, translate, store and retrieve multilingual information from rich multimedia documents in various languages.
HDecode	Cambridge University large vocabulary continuous speech recognizer
HLDA	heteroscedastic linear discriminate analysis
HMM	hidden markov model
HTK	hidden markov model toolkit
HTTP	The Hypertext Transfer Protocol (HTTP) is a networking protocol for distributed, collaborative, hypermedia information systems.
HUB4	1997 broadcast news corpus
Java	Java refers to a number of computer software products and specifications from Sun Microsystems that together provide a system for developing application software and deploying it in a cross-platform environment.
JavaScript	JavaScript is a script language typically used to enable programmatic access to computational objects within a host environment, commonly a web browser.
KAZN	a Mandarin commercial radio station
KLT	Karhunen Loéve transformation
LDA	linear discriminate analysis
LDC	Linguistic Data Consortium
LM	language model

Lucene	a high-performance, full-featured text search engine library written entirely in Java.
Machine Translation	Machine translation (MT) is a sub-field of computational linguistics that investigates the use of computer software to translate text or speech from one natural language to another.
MediaWiki	MediaWiki is a free software wiki package written in PHP, originally for use on Wikipedia.
Metadata	Metadata is data providing information about one or more other pieces of data.
MFCC	mel-frequency cepstral coefficient
MIT	Massachusetts Institute of Technology
ML	maximum likelihood
MLF	master label file
MLP	multi-layer perceptron
MPE	minimum phone error
MT	machine translation
MySQL	MySQL is a relational database management system (RDBMS) that runs as a server providing multi-user access to a number of databases.
NIST	National Institute of Standards and Technology
NASA	National Aeronautics and Space Administration
NEBULA	An open-source cloud computing project and service developed by NASA
Netflix	An American corporation that offers both on-demand video streaming over the internet in the United States and Canada, and flat rate DVD and Blu-ray Disc rental-by-mail in the United States.
NLP	natural language processing
NTDTV	New Tang Dynasty Television
OCR	optical character recognition
PDF	Portable Document Format (Adobe)
PDFBox	Apache PDFBox is an open source Java PDF library for working with PDF documents.
PER	phoneme error rate
Perl	Perl is a high-level, general-purpose, interpreted, dynamic programming language.
Perl DBI	The Perl DBI (DataBase Interface) offers a standardized way for programmers using the Perl programming language to embed database communication within their programs.
PHP	PHP: Hypertext Preprocessor is a widely used, general-purpose scripting language that was originally designed for web development to produce dynamic web pages.
PLP	perceptual linear prediction
Puppet	Puppet is an open source configuration management tool for managing the configuration of Unix-like systems declaratively.
Python	Python is an interpreted, general-purpose high-level programming language whose design philosophy emphasizes code readability.
RAID	RAID, an acronym for Redundant Array of Independent Disks (formerly Redundant Array of Inexpensive Disks), is a technology that provides

	increased storage reliability through redundancy, combining multiple relatively low-cost, less-reliable disk drives components into a logical unit where all drives in the array are interdependent.
RASTA	relative spectra
RDBMS	Relational DataBase Management System
RFA	Radio Free Asia
RT-03	2003 National Institute of Standards rich transcription evaluation
RT-04	2004 National Institute of Standards rich transcription evaluation
RWTH	Rhine-Westphalian Technical University
SAT	speaker adaptive training
SATA	Serial Advanced Technology Attachment (SATA) is a computer bus interface for connecting host bus adapters to mass storage devices such as hard disk drives and optical drives.
SAX	SAX (Simple API for XML) is a sequential access parser API for XML. SAX provides a mechanism for reading data from an XML document.
SCREAM	Speech and Communication Research, Engineering, Analysis, and Modeling
SCTK	National Institute of Standards speech recognition scoring toolkit
SDLT	Super Digital Linear Tape (SDLT) is a higher capacity version of Digital Linear Tape (DLT), a magnetic tape data storage technology developed by Digital Equipment Corporation.
SER	sentence error rate
SID	Speaker IDentification
SLM	statistical language modeling
Solr	Apache Solr is an open source enterprise search platform from the Apache Lucene project.
Solr CEL	Solr Content Extraction Library, a Solr feature that uses the content-extraction capabilities of Apache Tika to parse common office document formats.
SolrRequestHandler	A SolrRequestHandler is a Solr Plugin that defines the logic executed for any request.
SRILM	a language modeling toolkit developed by Stanford Research Institute
SPaRe	speech processing and recognition
Sphinx-4	an open source large vocabulary continuous speech recognition engine
SQL	Structured Query Language
SWAT	SCREAM Wikipedia Aided Translation
SYSTRAN-USG	Version(s) of Systran Machine Translation (MT) software for the US Government
TDT4	Topic Detection and Tracking corpus
TIFF	tagged image file format
Tika	A toolkit from the Apache Software Foundation for detecting and extracting metadata and structured text content from various documents using existing parser libraries
TRANSTAC	Translation System for Tactical Use corpus
UI	User Interface

Ultra-320	An implementation of the Small Computer System Interface (SCSI) standard with data transfer rates of 320 MB/s.
VTLN	vocal tract length normalization
VOA	Voice of America
VOIP	Voice over Internet Protocol (VOIP) is any of a family of methodologies, communication protocols, and transmission technologies for delivery of voice communications and multimedia sessions over Internet Protocol (IP) networks, such as the Internet.
WER	word error rate
Wikipedia	Wikipedia is a free, web-based, collaborative, multilingual encyclopedia project supported by the non-profit Wikimedia Foundation.
WPAFB	Wright-Patterson Air Force Base
WSJ	Wall Street Journal corpus
XHTML	XHTML (eXtensible HyperText Markup Language) is a family of XML markup languages that mirror or extend versions of the widely used Hypertext Markup Language (HTML), the language in which web pages are written.
XML	extensible markup language