

## Table of Contents

### Section 2 – Technical Details

<b>1</b>	<b>INNOVATIVE CLAIMS .....</b>	<b>1</b>
<b>2</b>	<b>PROPOSAL ROADMAP .....</b>	<b>3</b>
<b>3</b>	<b>TECHNICAL APPROACH .....</b>	<b>6</b>
3.1	INTRODUCTION .....	6
3.2	ACHIEVING ROBUSTNESS IN NOISE .....	6
3.2.1	Noise-robust Feature Extraction .....	6
3.2.2	Data-guided Trainable Feature Projection.....	7
3.2.3	Multi-resolution Tokenization.....	8
3.2.4	Noise Compensation .....	9
3.2.5	Fusion-based Classification.....	10
3.3	APPLICATION-SPECIFIC METHODS.....	11
3.3.1	Speech Activity Detection (SAD).....	11
3.3.2	Speaker Identification (SID).....	12
3.3.3	Language Identification (LID).....	12
3.3.4	Keyword Spotting (KWS).....	13
3.4	ACHIEVING THE PROGRAM GOALS .....	14
3.5	SOFTWARE INTEGRATION AND DELIVERIES.....	14
3.6	SPEECH TRANSCRIPTION OPTION .....	15
<b>4</b>	<b>COMPARISON WITH CURRENT TECHNOLOGY.....</b>	<b>16</b>
4.1	NOISE COMPENSATION.....	16
4.2	SPEECH ACTIVITY DETECTION .....	16
4.3	SPEAKER IDENTIFICATION .....	16
4.4	LANGUAGE IDENTIFICATION .....	17
4.5	KEYWORD SPOTTING.....	17
<b>5</b>	<b>STATEMENT OF WORK.....</b>	<b>18</b>
<b>6</b>	<b>INTELLECTUAL PROPERTY .....</b>	<b>28</b>
<b>7</b>	<b>SCHEDULE AND MILESTONES .....</b>	<b>29</b>
<b>8</b>	<b>PERSONNEL, QUALIFICATIONS, AND COMMITMENTS .....</b>	<b>30</b>
8.1	BBN .....	30
8.2	JOHNS HOPKINS UNIVERSITY.....	32
8.3	UNIVERSITY OF MARYLAND .....	32
8.4	CAMBRIDGE UNIVERSITY.....	33
8.5	BRNO UNIVERSITY OF TECHNOLOGY .....	34
8.6	TIME COMMITMENTS.....	35
8.7	SELECTED BIBLIOGRAPHIES .....	37
8.7.1	BBN .....	37
8.7.2	Johns Hopkins University .....	39
8.7.3	University of Maryland .....	41
8.7.4	Cambridge University.....	42
8.7.5	Brno University of Technology .....	43
<b>9</b>	<b>PROJECT MANAGEMENT AND INTERACTION PLAN .....</b>	<b>45</b>

9.1	PROJECT MANAGEMENT .....	45
9.2	TEAM SIZE AND COMPOSITION .....	46
9.3	WORKING/MEETING MODELS .....	47
9.4	SOFTWARE/CODE MANAGEMENT.....	48
9.4.1	Software Integration/Management .....	48
9.4.2	Secure Processing.....	49
9.5	UNIVERSITY PARTICIPATION .....	49
9.6	GOVERNMENT'S ROLE .....	50
9.7	PROJECT MANAGEMENT AT BBN .....	50
10	<b>COST SUMMARIES .....</b>	<b>51</b>
11	<b>ORGANIZATIONAL CONFLICT OF INTEREST AFFIRMATIONS AND DISCLOSURE .....</b>	<b>56</b>
12	<b>HUMAN USE .....</b>	<b>57</b>
13	<b>ANIMAL USE .....</b>	<b>58</b>
14	<b>STATEMENT OF UNIQUE CAPABILITY PROVIDED BY GOVERNMENT OR GOVERNMENT-FUNDED TEAM MEMBER.....</b>	<b>59</b>
15	<b>GOVERNMENT OR GOVERNMENT-FUNDED TEAM MEMBER ELIGIBILITY .....</b>	<b>60</b>
	<b>APPENDIX A: REFERENCES.....</b>	<b>61</b>
	<b>APPENDIX B: EFFECT OF TARGET SIZE .....</b>	<b>66</b>

# RATS PATROL

---

## Volume 1 – Technical Proposal Section II – Technical Details

### Technical Area 1: Algorithmic Development and Signal Processing





## 1 Innovative Claims

### The Challenge

The ability of the US Government to monitor foreign voice communication channels, in support of intelligence and military operations, is hampered by the lack of a technology that can automatically determine what is important to listen to in the presence of highly degraded and noisy voice signals. Developing such a technology is the aim of the RATS program, and BBN's response is PATROL, a system that will revolutionize US ability to pinpoint speech activity, identify the language and the speaker, and detect keywords of interest, under very noisy and degraded conditions.

### Barriers to Progress

Two major barriers to dealing with noisy speech have been fragmentation of research efforts and lack of a realistic corpus on which to perform research. The RATS program promises to shatter these barriers by attacking simultaneously the full spectrum of speech detection problems and by providing the first large realistic corpus for research. BBN has responded to the challenge by forming a team whose members have distinguished themselves internationally in their areas of expertise and who collectively have the breadth and depth of experience, history of innovation, and fresh new ideas to meet the RATS challenge in all areas of this effort.

### The Approach

Our approach derives from a holistic view of the problem at hand as a single overarching

challenge of speech detection/recognition in noise. The image we have of noisy speech is that the speech is being held hostage by noise. Our fundamental approach takes advantage of the *predictable* properties of the speech to rescue it from the clutches of *unpredictable* noise. PATROL will accomplish the rescue through a *unified* multi-pronged attack that is common to all four RATS application areas: speech activity detection (SAD), language ID (LID), speaker ID (SID), and keyword spotting (KWS). Figure 1-1 shows a conceptual architecture of the PATROL system which combines elements of training and recognition in a single notional diagram. The double arrows indicate multiple analyses and outputs. The terms "data-guided" and "task-specific" will then be specific to the available training and the application area. We now describe each component briefly.

**1. Noise-Robust Feature Extraction:** We propose the use of auditory-based representations of sound that have been demonstrated to separate speech modulations from the noise in a higher-dimensional space, which should make it possible to focus our processing on the speech modulations. These representations include Hermansky's *modulation spectrum* and Shamma's 4-D *cortical spectrum*. (See also item 4 below.)

**2. Data-Guided Feature Projection:** To make the higher-dimensional features usable, we propose to project them onto a lower-dimensional subspace – with little loss in separability of speech from noise – by using non-linear, trainable, discriminative



Figure 1-1 Conceptual PATROL architecture for all four applications: SAD, LID, SID, KWS.



projections based on multi-layer perceptrons and region-dependent transforms. One particularly useful projection that has shown very good performance in noise is the estimation of phoneme posterior probabilities as features. In all cases, the training is guided by the available data and the application.

**3. Multi-resolution Tokenization:** More than anything, what characterizes speech – and differentiates it from noise – is the special manner in which speech units (tokens) vary over time. We propose to model speech as a time sequence of tokens at different time resolutions – frames, phoneme-like *self-organized units*, phonemes, and words. To model the time sequence of these tokens, we shall employ language models that are appropriate for the tokens and the application. The resulting output here is multiple lattices of tokens, with associated likelihood ratios (scores), to be used for later decision making (item 5 below). We also introduce methods that do not require extensive transcription.

**4. Noise Compensation:** Noise compensation methods improve recognition accuracy significantly. Two types of noise compensation will be employed. One, at the feature level, performs discriminative training of the features to minimize the effect of the noise. Another, model-based, is performed in conjunction with tokenization. These methods, which use discriminative adaptive training, include Vector Taylor Series compensation, Joint Uncertainty Decoding, and Predictive Constrained Maximum Likelihood Linear Regression. Joint Factor Analysis, which has been very useful in modeling speaker and channel variability, will be used for noise compensation as well.

**5. Fusion-based Classification:** The basic concept here is to fuse the outputs of multiple processes to reduce uncertainty in the presence of noise. Another is the fusion of multiple lattices of tokens and associated scores, at different resolutions, to formulate

an integrated, optimal decision. Different methods of fusion will be considered. One is based on the Dempster-Shafer theory of evidence. Another novel method is a condition-adaptive fusion strategy, called *latent generalized linear model*, in which combination weights of evidence are functions of measured features of the input, such as SNR, type of channel, and speaking rate. Another aspect of fusion is a principled approach to determining an optimal threshold for acceptance/rejection.

**6. Model Adaptation:** The high-confidence fusion decisions can then be used to adapt the speech token models (item 3 above) in a feedback configuration to improve accuracy.

**7. Task-specific Discriminative Training:** Permeating the above components is the use of discriminative training, whereby the various components are trained specifically for each application to optimize performance.

### Basis of Confidence

While some of the above ideas have been used individually to combat noise, what we propose is an integrated approach that combines *all* the components in a novel and unified manner for optimal performance. In Section 3, we present our novel ideas in how to optimize performance for each of the application areas (SAD, LID, SID, KWS), but the centerpiece of this proposal is the unified system conception outlined above.

The experienced, world-class team we have assembled; the large number of fresh ideas and unifying new methodology in this proposal; and BBN's track record in fielding operational systems, present a uniquely compelling and credible opportunity for achieving the ambitious performance goals of the program. We are confident that we will deliver to the Government deployable technology that will combat the endemic noise problem, rendering our monitoring operations far more effective.



## 2 Proposal Roadmap

AREA	Page References
<p><b>a. Main Goals of the Proposed Research:</b></p> <p>The work in this project has two main goals:</p> <p>(1) The first goal is to develop the technology to perform the following four speech detection tasks, for a number of foreign languages, in the presence of highly degraded and noisy voice signals, with high accuracy: Speech activity detection (SAD), language identification (LID), speaker identification (SID), and keyword spotting (KWS). The miss and false alarm target rates for the four applications are given in Table 3-1. The work will include the delivery of an evaluation system to the Evaluation Team in order to assess system performance against the targets for each phase of the project.</p> <p>(2) The second goal is to build and deliver to the Government a system that is military user-friendly, with an intuitive, easy-to-use graphical user interface (GUI), and that will be field-trainable for new languages, speakers, and keywords.</p>	<p>1 (§1) 14-15 (§3.4, 3.5)</p>
<p><b>b. Tangible Benefits to End Users:</b></p> <p>By providing the Government with the capabilities described above, with high accuracy, we would increase the ability of the Government to monitor foreign communications significantly. By making it possible for the analysts to focus only on the languages, speakers, and keywords of interest, the productivity of the analysts would increase tremendously.</p>	<p>1-2 (§1)</p>
<p><b>c. Critical Technical Barriers:</b></p> <p>(1) While much work has gone into the four RATS application areas, much of that work has been on relatively clean speech. As a result, the techniques that have been developed do not perform well under high noise conditions.</p> <p>(2) For cases where research was done with noisy signals, typically the research focused on one technique at a time – rarely combining multiple techniques.</p> <p>(3) The research is often done using data with artificially added noise, rather than with naturally occurring noisy speech. Any naturally occurring noisy speech is usually collected in one type of environment, e.g., in cars, and the tasks are usually very narrow, such as digit recognition.</p> <p>The RATS program is well poised to overcome these barriers by attacking the full spectrum of speech detection problems and by providing the first large realistic corpus for research.</p>	<p>6 (§3.1) 16-17 (§4)</p>
<p><b>d. Main Elements of the Proposed Technical Approach:</b></p> <p>We propose to build a system, PATROL, which will achieve the goals of the program through a <i>unified</i> approach to detection systems that are robust to noise. Instead of the piecemeal, one-technique-at-a-time approach of many past efforts, many techniques at multiple levels of processing will be developed and optimized jointly to maximize performance for each of the RATS application areas. The primary areas of work include:</p> <p>(1) <b>Noise-robust feature extraction:</b> It has been shown that, when noisy speech</p>	<p>1-2 (§1)  6-7 (§3.2.1)</p>



is analyzed using multi-dimensional (multi-D), auditory-based representations, the modulation spectrum of speech typically occupies a different part of the space than the noise. We plan to use several such representations jointly, thus leading to increased robustness, including the multi-D MRASTA modulation spectrum of Hermansky and the cortical representation of Shamma (both on the PATROL team).	
(2) <b>Data-guided trainable feature projection:</b> We propose to use nonlinear discriminative projections based on multi-layer perceptrons and region-dependent transforms (RDT) to reduce the dimensionality of these high-dimensional features with minimal loss of information. The term “data-guided” means that these projections will be optimized to preserve information needed for each application (SAD, LID, SID, or KWS).	7-8 (§3.2.2)
(3) <b>Multi-resolution tokenization:</b> Speech can be analyzed as tokens (units) of different length, e.g., frames, phonemes, sub-words, or words. In contrast to current methods, we propose to use all levels of tokenization, which will increase the robustness of all applications. We also propose two methods that have been demonstrated to work well with untranscribed speech (which is abundant in an operational setting): <i>self-organized units</i> (SOU) and semi-supervised training.	8-9 (§3.2.3)
(4) <b>Noise compensation:</b> Among <i>feature-based</i> methods, we propose to filter the multi-D features to enhance the speech and suppress the noise, and to explore discriminative RDT to map from noisy features to clean features. Among <i>model-based</i> methods, we will continue to explore sophisticated techniques pioneered by Cambridge University (e.g., Joint Uncertainty Decoding and Predictive Constrained Maximum Likelihood Linear Regression), and Joint Factor Analysis – which has been used effectively at Brno University of Technology to deal with inter-session variability.	9-10 (§3.2.4)
(5) <b>Fusion-based classification:</b> The various types of analysis and tokenization will result in a number of lattices of tokens with scores, which will then be combined using <i>condition adaptive fusion</i> . For SID and KWS, decisions will be optimized for <i>sets</i> of targets rather than for individual targets.	10 (§3.2.5)
(6) <b>Model adaptation:</b> The high-confidence fusion decisions can be used to adapt the speech token models (item 3 above) in a feedback configuration to improve detection accuracy.	10 (§3.2.5)
(7) <b>Task-specific discriminative training:</b> Permeating the above components is the use of discriminative training, whereby the various components are trained specifically for each application to optimize performance.	6-14 (§3.2, 3.3)
(8) <b>Speech activity detection (SAD):</b> In addition to using noise compensation methods, we expect SAD to improve through the use of phoneme and SOU HMMs, as well as via an integrated factorial HMM speech/noise recognizer.	11 (§3.3.1)
(9) <b>Language identification (LID):</b> LID will be more accurate with the use of more detailed SOUs, phoneme, and word models to compute acoustic likelihood ratios, and with the combination of scores across multiple levels of tokenization. Speaker Adaptive Training will reduce the speaker variation effect on LID.	12-13 (§3.3.3)
(10) <b>Speaker identification (SID):</b> SID will also improve through the use of all	12 (§3.3.2)



levels of tokenization for acoustic likelihood ratios and sequence probabilities. These scores will be fused with other classifiers such as SVMs that use super vector representations (e.g., MLLR parameters or mean vectors) of each speaker.	
(11) <b>Keyword spotting (KWS):</b> Our KWS system will be based on large-vocabulary speech recognition to model both keywords and other speech. We propose to use higher-level language information, such as topic-dependent language models, and unsupervised adaptation based on high-confidence hits to optimize performance.	13-14 (\$3.3.4)
(12) <b>Software integration and deliveries:</b> BBN will use its vast experience in integrating software from other sites and delivering real-time, operational systems to the Government to build and deliver two systems: an <i>evaluation system</i> to be delivered to the Evaluation Team to assess system performance against program targets, and a <i>trainable system</i> as described in (a) above. The evaluation system will be delivered six weeks prior to the end of each phase, and the trainable system will be delivered in Phases 2 and 3.	14-15 (\$3.5)
(13) <b>Speech transcription option:</b> We have found that KWS users prefer to have a full transcription – even if errorful – to avoid having to listen to every possible hit. In order to fulfill the promise of the ‘T’ in ‘RATS’, we have included work on full transcription as an option in Phases 2 and 3.	15 (\$3.6)
<b>e. Basis of Confidence:</b> In short, two things: The approach and the team. Noise is the main problem and our whole approach is based on various principled ways of dealing with it. We have great confidence that the unified approach presented here and the specific set of novel techniques that we plan to use will, in combination, deal very effectively with the problem of noise in speech. To perform this challenging work, we will employ the combined talents of an experienced team with a strong history of achievement in every aspect of the program. The existing strong relationships among members of the team will enhance our ability to perform. BBN’s experience in executing multi-site speech and language projects – always with superior results – and our proven record of integrating software and deploying real-time, operational systems for the Government, gives us additional confidence that we will be able to provide the analysts with what they need to be more effective in monitoring communications.	1-2 (\$1) 6-15 (\$3) 30-35 (\$8.1-8.5) 45-50 (\$9)
<b>f. Nature and Description of End Results to be Delivered to DARPA:</b> Six weeks before the end of each Phase, BBN will deliver an <i>evaluation system</i> to the Evaluation Team. A <i>trainable system</i> will also be delivered to the Government during Phases 2 and 3. The PATROL team will also engage in the writing of technical papers and conference presentations which will detail the approaches being taken, the novel ideas that have been explored, and the results of carefully controlled experiments to assess the benefits of competing methods.	14-15 (\$3.5) 18-19, 22-23, 25-26 (\$5) 29 (\$7)
<b>g. Cost and Schedule of the Proposed Effort:</b> Base Phase 1: \$5,293,372; Base Phase 2: \$3,895,062; Base Phase 3: \$3,894,111; Option Phase 2: \$482,785; Option Phase 3: \$499,559.	51-55 (\$10)



### 3 Technical Approach

#### 3.1 Introduction

Much of the research in the four applications in the RATS program (SAD, LID, SID, KWS), has been done either under relatively clean speech conditions, or by artificially adding different types of stationary noise to the speech. In addition, existing research typically applies a single technique at a time to deal with the noise, rarely taking advantage of multiple methods to improve performance.

The reality for conditions of interest to the Government is quite different. The noise is typically neither simply additive nor is it stationary, and can be quite severe. Under such conditions, the current technology is inadequate for the applications of interest.

We are fortunate that the RATS program will be providing us, for the first time, the types of realistic communications noisy speech data on which to perform the research. The program has also given us the opportunity to work on the four detection problems with such data simultaneously, and to partner with other experts to solve this very challenging problem.

Our proposed PATROL system, depicted in Figure 1-1, will employ multiple noise-robust solutions across all four RATS applications in a unified way: features that exploit the distinctive spectral and temporal modulation patterns of speech and noise; acoustic and language models that capture contextual information; adaptive noise compensation in both feature and model space; multi-stream information fusion; and model adaptation. PATROL will use these solutions and the available RATS data, in combination with data-guided and discriminative training techniques, to optimize performance for each of the four RATS applications.

Our general techniques for robustness in noise are described in Section 3.2. Section 3.3 presents methods that are specific to each of the four RATS applications. A timeline for

meeting the goals of the RATS program is provided in Section 3.4. Software integration and delivery to the Government are discussed in Section 3.5. Finally, in Section 3.6, we propose an option for automatic transcription of noisy speech. The numbered references below are provided in Appendix A of this proposal.

#### 3.2 Achieving Robustness in Noise

Our approach to achieving robustness in noise is presented below under five headings: 1) noise-robust feature extraction; 2) data-guided feature projection; 3) multi-resolution tokenization; (4) noise compensation; and 5) fusion-based classification (including model adaptation). In Section 3.3, we show how these techniques apply to each of the four RATS application areas.

##### 3.2.1 Noise-robust Feature Extraction

Our attack on noise starts at the front-end, feature extraction. MFCC [1], PLP [2], and RASTA [3] features, along with their time derivatives, the use of cepstral mean subtraction and/or spectral noise subtraction [4], form an initial line of attack against which we will compare more sophisticated noise-robust methods. Prosodic features (pitch envelope, probability of voicing) have also been found useful in LID and SID.

All successful feature extraction techniques are based on some aspect of our knowledge of human auditory perception. In this effort, we plan to explore a series of progressively more sophisticated uses of that knowledge. We start with Hermansky's Frequency Domain Perceptual Linear Prediction (FDPLP) [5], a "plug-in compatible" alternative to PLP that has been shown to provide more noise robustness by focusing attention on the high SNR regions of the signal.

Auditory experiments over the last decade suggest that humans perceive sound in a *multi-dimensional* (multi-D) space consisting of responses of neurons that operate on the



spectrogram of the sound and selectively extract its important spectral and temporal trajectories, or *modulations*. It appears that, in this multi-D space, speech and noise typically occupy different regions of the space, due to their different modulation patterns. One example of these multi-D representations is the MRASTA *modulation spectrum* from Hermansky [6], which extracts, at each frequency and time, vectors of information over longer time and frequency spans.

A more sophisticated multi-D processing is the *cortical representation* of Shamma and group [7], which analyzes the spectro-temporal modulations of the auditory spectrogram by applying a 2-D affine wavelet transform on the spectrogram, with spectral dilations over a range of bandwidths (or *scales*) of 0.25-8 cycles/octave and time dilation *rates* of 1-32 Hz. The output is a set of scale vs. rate matrices – the cortical spectrum (CS) – at *each* point in frequency and time. The process is applied at every 10 ms of audio, yielding a time series of 3-D (frequency/rate/scale) tensors.

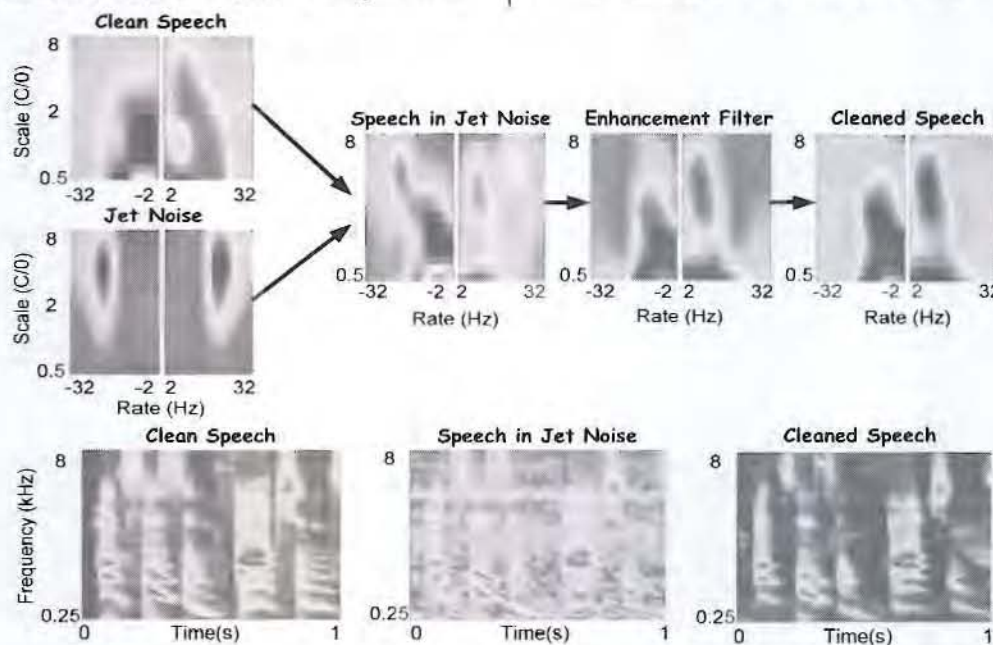
To illustrate the potential of the CS in terms of robustness to noise, consider the speech-in-

jet-noise example in Figure 3-1. Shown on the top left are the CS for clean speech and jet noise, *averaged* for all time and frequency points over a 1-second span of the signal.

This CS clearly shows how jet noise largely occupies a different part of the CS space than does speech, while in the speech-in-jet-noise ordinary spectrogram below, the two spectra overlap. The figure then shows the CS of an enhancement or correction filter which, when applied to the speech-in-jet-noise CS, yields a cleaned-speech CS that resembles the original CS of the clean speech [8]. The resulting CS (now *without* averaging over frequency and time) will form our multi-D feature representation for further processing.

### 3.2.2 Data-guided Trainable Feature Projection

The above feature extraction methods produce a multi-D feature vector (~7,000 dimensions) at every 10 ms of speech! So, modeling the resulting features efficiently and reliably requires projecting them to a sub-space of significantly lower dimensionality, without losing the speech/noise separability property. Standard linear dimensionality reduction techniques, such as principal component



**Figure 3-1: Example of noise compensation in cortical representation (top row). The bottom row shows the corresponding auditory spectrogram.**



analysis (PCA) or linear discriminant analysis (LDA), can be useful but are not powerful enough for the task at hand. We propose to investigate *non-linear discriminative* projections based on multi-layer perceptrons (MLP), and the region-dependent transforms (RDT) [9] developed at BBN.

Posterior-based MLPs map the input features to vectors of phoneme posterior probabilities. The MLP parameters are estimated on a phoneme-labeled corpus to minimize frame-level phoneme classification error. This method has been shown to yield significant increase in recognition accuracy, especially under noisy conditions [10] [11], with further improvements possible by training the MLPs in a hierarchical manner [12] [13].

The RDT partitions the input feature space into regions, each with its own projection. Given an input feature vector, RDT computes the probability of each region using a partition model, which can be customized for different types of parameters. For example, a region could correspond to a single Gaussian in a Gaussian Mixture Model (GMM), or a phoneme, or even a state in a hidden Markov model (HMM). When the estimation of the transforms in RDT is performed discriminatively, the resulting projections have been shown to give significant speech recognition accuracy improvements [14].

Both MLPs and RDT are *trainable, data-guided* feature projections. Given an annotated training corpus and the associated raw features, they can learn to find the feature dimensions that make the desired distinctions that are least affected by noise. Here we refer to distinctions that are most important for the task of interest, such as speech/nonspeech in SAD, speakers in SID, and languages in LID. In KWS, MLP and RDT transforms can be trained to minimize frame phoneme classification error or expected phoneme error rate, respectively. Of the two types of discriminative non-linear projections, RDT

may have the larger potential for success in this project, due to its ability to accommodate complex classifiers, such as HMMs.

### 3.2.3 Multi-resolution Tokenization

*Tokenization* is the process of converting the speech signal into a sequence of tokens, and *multi-resolution* refers to the tokens being of different time resolutions (e.g., frames, phonemes, words). We propose to model the special variations of these tokens over time to re-capture the speech from the noise. Multi-resolution models will be estimated for each of the four RATS applications.

#### Modeling Sequences over Time

To break the noise barrier in SAD, we must go beyond the use of time-agnostic GMMs and use the fact that noisy speech still consists of a time sequence of sounds, at a predictable rate of about 6-10 phonemes per second. We propose to discriminate speech from non-speech by modeling the sequence of speech tokens in time explicitly. For SAD to be language-independent, we would limit the resolution of the tokens up to the phone level.

LID and SID systems already model speech sequences using a tokenizer (typically a generic phoneme recognizer), and they model the likelihood of those sequences using an n-gram model. Here, we also propose to use tokenizers to model the *acoustics* for the target language or speaker, so that differences in pronunciation can be captured automatically. The acoustic likelihoods will be measured for tokens at different time resolutions. Each type of token has its advantages and disadvantages, and the choice of tokenization will be influenced by the amount of training data and the type of data annotations available. For example, we may not always have substantial amounts of transcribed training speech for training word-based recognizers. We propose two novel solutions for the case where transcriptions are not available: semi-supervised training and Self-Organized Units (SOU).



### Semi-Supervised Training

We have shown that, given large amounts of un-transcribed audio, in conjunction with a small amount of transcribed speech (e.g., one hour), we can achieve speech recognition accuracy that approaches that of having manual transcripts for the larger corpus [15][16][17]. Starting with a model estimated from the small amount of transcribed speech, the method uses iterative re-estimation of the model based on recognizing the untranscribed audio, coupled with appropriate confidence filtering. Recently, the method has been shown to work on noisy speech data as well. We propose to use semi-supervised training with the RATS data, as well as any other untranscribed audio data we can find in the five languages of interest, to improve tokenization accuracy. Note that this method can gain special importance in an operational setting, where essentially unbounded amounts of untranscribed data are typically available.

### Self-Organized Units (SOUs)

We have developed another technique that can take advantage of untranscribed audio when we have *no* transcribed speech available, or even a vocabulary for the language [18]. The method starts by dividing the audio into phoneme-like units using a simple spectral change criterion. These units are clustered to produce initial pseudo-phone models, called *self-organized units* (SOUs), which are then re-estimated using context-dependent HMMs. We can use SOUs to tokenize both training and test data. For SAD, SOUs can be used to model both speech and nonspeech segments. An acoustic model and a language model of SOU sequences for different languages can be used in LID. For SID, we can find the sequences of SOUs that are frequently produced by each speaker.

#### 3.2.4 Noise Compensation

We group noise compensation techniques into two categories: (a) *feature-based* methods, which process acoustic features in some

fashion to suppress the noise, and (b) *model-based* methods that alter the parameters of the acoustic model based on some estimate of any residual noise in the features.

#### Feature-based Noise Compensation

Since the modulation and cortical spectra, presented in Section 3.2.1, attempt to separate the speech from the noise in multi-D, we propose to introduce filters in those domains to compute new features that enhance the speech and suppress the noise. Shamma has already demonstrated the utility of this approach in SAD [19]. We propose to generalize the approach to other applications.

We will also explore *RDT-based enhancement*, a new method that relies on discriminatively estimated region-dependent transforms (see Section 3.2.2) to perform a mapping from noisy features to clean features. Unlike other feature mapping techniques, such as SPLICE [20], which require stereo (noise/clean) speech samples for training, our method only requires a noisy training corpus, and a target acoustic model of clean speech. The clean speech model may be obtained from any existing clean speech corpus, not necessarily of the same spoken language.

#### Model-based Noise Compensation

Sophisticated model-based compensation techniques, such as those pioneered by our partners at Cambridge University [21][22], have yielded excellent recognition results, even in low SNR. These approaches can also be used within an adaptive training environment allowing the estimation of a neutral, canonical, speech model that is suited for adaptation to a range of target environments. Specifically, this will include improving existing work on discriminative adaptive training approaches based on schemes such as Vector Taylor Series compensation [23], Joint Uncertainty Decoding (JUD) [24], and Predictive Constrained Maximum Likelihood Linear Regression (PCMLLR) [25]. Furthermore,



we will investigate methods that combine model-based noise compensation with unsupervised speaker adaptation, such as Noisy CMLLR adaptation [26].

Another technique that has been used extensively in recent SID and LID systems to deal with inter-session variability is joint factor analysis (JFA) [28]. This method models the effect of speaker and channel variability on the parameters of the acoustic model used in such applications, typically a GMM. JFA can be augmented to address variability due to noise as well, and can be applied to more complex acoustic models, such as HMMs, by generalizing the form of the transformations on the model parameters. Our partners at Brno University of Technology (BUT) will explore these research directions.

### 3.2.5 Fusion-based Classification

Our proposed system will use multiple feature analyses and models, as described in the previous sections, whose decisions – in the form of lattices of speech tokens with scores – will be ultimately combined, or *fused*, in order to generate the final system output for each of the four RATS applications.

A step in this direction is fusion based on the Dempster-Shafer theory of evidence [29]. This strategy suppresses streams with high entropy of the classifier output, and enhances low-entropy streams. Our partners at Johns Hopkins University (JHU) have applied this type of fusion on the phonetic posteriors from multiple feature streams [30], and are currently looking into alternative confidence measures for combining the independent stream decisions.

An alternative method for combining different classifiers is by directly optimizing the fusion model on a held-out set. We propose to extend this approach using a condition-adaptive fusion strategy, called *latent generalized linear model* (LGLM), in which the combination weights are functions of

measured features of the input, such as SNR, channel type, and speaking rate. The LGLM parameters can be estimated to explicitly optimize performance in each application.

### Optimal Decisions and Target Set Size

In 1986, while working on a BBN-funded speaker verification project [31], we found it best to use a different threshold for each target speaker, or conversely, normalize the scores differently for each target so that a single threshold would result in the same false alarm (FA) rate for all targets. This was done by computing the distribution of scores for a large number of impostor samples against each target model. This technique is now commonly used for speaker spotting [32]. We have applied it to topic spotting [33][34], and a modified version more recently to KWS [35]. This method is of special significance because of the importance of controlling the FA rate in RATS.

The evaluation plan for SID and KWS in the RATS BAA calls for testing on a set of  $N$  targets at a time ( $N=10$  for SID and  $N=25$  for KWS), instead of the usual one target at a time. For the task of detecting *sets* of targets, we will modify the decision criterion to use all of the targets jointly to decide whether to accept or reject a test sample. In principle, if we know this set of targets during model training, we can train all of the models discriminatively to directly estimate the posterior probability of the target set. Alternatively, we can use this information in the fusion classifier, which can estimate directly the posterior probability of the target set given all of the scores from the different models and targets.

### Model Adaptation

Fusion decisions will include confidence scores. For each application, we propose to use high-confidence hits to adapt the speech models (tokenizers as well as acoustic scorers) and re-process the test data to improve detection accuracy.



### 3.3 Application-Specific Methods

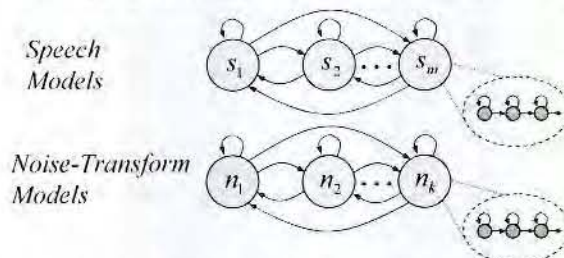
Here we summarize the techniques we will use for each of the four RATS applications. In all four applications, we shall assume that, at a minimum, we start with the noise-robust feature extraction techniques in Section 3.2.1, at increasing levels of complexity, up to the modulation and cortical spectra, which promise to segregate the speech and the noise in a multi-D space, followed by the feature-based noise compensation techniques in Section 3.2.4. Then, when necessary, we will use the data-guided feature projection techniques in Section 3.2.2 to reduce the feature dimensionality. This step will be performed discriminatively, based on the available training data and the application. The result will be what we will call below the *processed features*, or simply features. The types of tokenization (Section 3.2.3) to be used will depend on the application.

#### 3.3.1 Speech Activity Detection (SAD)

The processed features will be used to discriminatively train both static frame-based classifiers (e.g., GMM, or Support Vector Machines – SVM), as well as phoneme-like HMM tokenizers. Initially, we will explore HMM tokenizers configured as we have used for SAD in broadcast news, where speech is often corrupted by background noise or music. In that work [36], we used an HMM consisting of phoneme-class tokens (e.g., nasal, fricative, vowel), as well as non-speech tokens (e.g., music, laughter, background noise). We will also investigate SOU-based HMM tokenizers, with one set of SOUs for noisy speech and another set for pure noise. A simplified version of this tokenizer, using just two SOUs (speech/non-speech), was recently applied with success to air traffic control audio data [37]. Both of these types of HMM tokenizers are expected to provide a good baseline for SAD on RATS data.

We also propose a bold novel approach that attempts to model the very process of

generating noisy speech through the composition of two *hidden* stochastic processes: a clean speech process and a *noise-transform* process. The new model takes into account the fact that noise is unpredictable and differs between training and test. By factoring the noisy signal into these two processes, the proposed model will be able to better generalize to unseen combinations of speech and noise. The proposed factorial HMM [27] tokenizer is depicted in Figure 3-2.



**Figure 3-2: A factorial HMM recognizer proposed for SAD.**

The factorial HMM recognizer consists of two parallel ergodic HMMs: an ordinary phoneme-like speech HMM (that also includes a special pause phoneme) and a novel noise-transform (or *noise-x*) HMM. These are two-level HMMs, with each state of the top level comprising a second-level HMM. The speech model is estimated from any existing corpus of clean speech data (the corpus can be in any language). The probability density functions (pdf) of the noise-x HMM are distributions over *transformations* that transform the speech model parameters into those of the noisy speech. (There will be a few noise-x HMMs, one for each expected type of noise.) Given the clean-speech HMMs, the parameters of the noise-x pdfs are estimated on RATS data. Since the noise-x transforms are state specific, the model can handle non-stationary noise. Note that the speech models are quite generic here and are for rather broad classes of sounds. The action is really in the noise-x HMMs which are trained to transform the speech models into the noisy speech models from the



RATS data. SAD is performed by finding the highest scoring sequence of factorial HMM states (pairs of noise-x and phoneme/pause states) given the noisy observations.

The above SAD methods will be combined or fused with that developed at the University of Maryland (UMD) using adaptive fusion (see Section 3.2.5) to optimize performance.

### 3.3.2 *Speaker Identification (SID)*

For accurate speaker identification, it is often necessary to not only determine whom the speaker sounds like (vocal tract characteristics, speaking rate, pitch contour, accent), but also whether certain pause fillers or distinguishing words or phrases are used. Relying on such *high-level information* becomes even more critical when the acoustic signal is degraded by noise or channel distortions.

Current state-of-the-art SID systems use simple models (e.g. GMMs) for modeling speaker vocal tract characteristics, and, in some cases, more complex automated speech recognition (ASR) models for deriving some of the higher-level information [38] [39]. The ASR models are trained in a typical, speaker-independent, channel-independent fashion, and are adapted to the speaker during testing using standard methods such as maximum likelihood linear regression (MLLR) [40]. However, the adaptation is estimated to optimize an ASR criterion, *not* to help in speaker identification. We believe there is a better way. Here we propose to *couple* speaker and speech recognition in a novel SID approach we call *speaker-discriminative ASR*, or SD-ASR. This approach also promises to perform especially well in noise.

Under SD-ASR, each speaker is modeled by a full ASR sub-system, with speaker-specific acoustic models derived via adaptation from a universal background acoustic model. What distinguishes SD-ASR from typical ASR is the fact that the parameters of the speaker-specific and background ASR models are estimated *jointly* so as to directly optimize

SID performance on training data. The training criterion is closely related to MMIE (maximum mutual information estimation) as developed for ASR, but it aims at maximizing the posterior probability of the correct speaker over the training data, rather than maximizing the posterior of the correct word sequence. The criterion can also be easily modified to optimize parameters for the scenario in which the target is not a single speaker, but rather a set of high value individuals (HIVs).

The proposed SD-ASR system can be used even in cases where little or no transcriptions are available, by applying semi-supervised training, or using self-organizing units as explained in Section 3.2.4. By combining multiple SD-ASR systems – each trained with different tokenizations (words, sub-word units, phonemes, SOUs) –with complementary state-of-the-art SID systems from our Brno University of Technology (BUT) and JHU partners via adaptive fusion, significant improvements in SID performance will be achieved.

### 3.3.3 *Language Identification (LID)*

As in SID, accurate LID must rely on multiple types of information extracted from the acoustic signal. But, unlike SID, LID needs to factor out differences due to speakers. For that purpose, standard approaches to LID make use of speaker-independent acoustic models to recognize lattices of phonemes or sub-word units, which are then rescored by multiple language-specific acoustic models and n-gram phonotactic models [41] to find the highest scoring language on a given segment of speech. The language-specific acoustic models aim at capturing distinctive general acoustic variations, such as pronunciation, prosody, or phoneme duration that are characteristic of the language being spoken. However, the acoustic models used in state of the art LID systems are typically GMMs [42], and hence have limited modeling capabilities. To strengthen the role of the



acoustic model and improve performance, we propose to introduce the use of highly accurate HMMs, trained discriminatively on the noise-robust features and with the model-based noise compensation approaches described in Section 3.2.4.

To ensure that our LID acoustic models are not confused by speaker differences, we will apply speaker-adaptive training (SAT) [43], which tries to map the training speakers for the language to the same compact feature space. Then, when classifying a new sample, we adapt each of these compact acoustic models to the test speaker before computing the likelihood. This way, the acoustic models will be less sensitive to speaker differences within a language.

One issue with n-gram phonotactic models is that they rely on local context and hence might be less reliable under noisy conditions due to inaccuracies in recognizing the correct sequence of speech tokens. We propose to alleviate this problem by applying information retrieval techniques, as in [44], which attempt to model the global distribution of token n-grams.

The final LID system output will be based on adaptively fusing the decisions of multiple classifiers (GMMs, HMMs, enhanced phonotactic models) applied to multiple tokenizations of the speech input, and incorporating the LID system from BUT.

### 3.3.4 Keyword Spotting (KWS)

In order to perform KWS, one must have a good model of the non-keywords as well as the keywords. It is now acknowledged that the best model for the non-keywords is a set of models for those words – both acoustic and language models (LM). This is especially true in the presence of noise, where making use of context gives us a better chance of detecting the keywords. Another benefit of having good models for the non-keywords is that they can then all serve as possible keywords, since it would not be possible to

know in advance what the keywords might be in the field (the case of unknown words is discussed below). BBN has developed state-of-the-art systems in both large-vocabulary speech recognition and KWS. Here, we propose extensions to that work.

One way of making better use of context, and to improve KWS performance in noise, is to use a topic-weighted LM. Since topics are typically not annotated in data, we propose to use our Unsupervised Topic Discovery (UTD) method [45] to discover a large set of topics for the language and the domain in an unsupervised manner.

Given the set of topics, we estimate from training the probability of each keyword occurring in a passage given that the passage contains each topic. Then, given a test passage, we first recognize it automatically, and then determine the likely set of topics contained in the passage. Using the topic probabilities, we compute modified prior probabilities for each keyword and then scale all the n-gram probabilities for this keyword to produce new priors. Finally, we perform a second pass of recognition – or just rescore the lattice of possible answers with this modified language model – in order to obtain an improved result.

Another method to improve KWS performance is to combine or fuse decisions from multiple independent KWS processes. Specifically, we will have a system that first runs recognition to generate a lattice of tokens (phonemes, sub-words, words), and then rescores the lattice with different acoustic models, based on HMMs or word-specific SVMs. The final adaptive fusion stage would take in the scores from the HMM and SVM classifiers, along with any additional measured features of the input in order to produce the final decision.

### Handling Keywords Outside the Training

So far we have assumed that the keywords are words that are already in the system's



vocabulary. The case of new words, unknown to the system, must be handled properly in an operational setting, including new names. We assume here that the words are provided to the system in the form of text. For the languages under consideration in the RATS program, having the spelling of a word is sufficient for the system to deduce a pronunciation for the word and, therefore, provide a reasonable initial acoustic model, coupled with an initial back-off LM.

The new keywords will have lower recognition accuracy at first. We propose to adapt the models for each keyword based on the high-confidence hits for that keyword. This includes the parameters of the context-dependent HMMs, the pronunciation probabilities, and the LM for these words. To improve the LM, we find those other words that often occur in the same observed context as these high-confidence tokens. Then, we replace the initial back-off LM probabilities for the keyword with a weighted interpolation of all of the LM probabilities relevant to those similar words, in proportion to their similarity and the confidence of the first recognition pass. Finally, with a complete set of new models for these keywords, we perform a second pass of recognition over the data set in order to obtain more accurate estimates of the locations of the new keywords.

### 3.4 Achieving the Program Goals

Table 3-1 shows our proposed performance timeline for the RATS effort over the three phases of the project. The baseline numbers for SAD, LID, and KWS were obtained in a DARPA-funded joint effort between BBN and MIT Lincoln Lab (MITLL) to assess the state of the art when applied to government-supplied, noisy communication speech data in four languages: Iraqi Arabic, Farsi, Dari, and Pashto. The baseline numbers for SID were supplied by MITLL from a previous study with noisy speech data. The Phase 3 targets are those specified in the BAA and the

interim targets were computed through logarithmic interpolation between the baseline and the Phase 3 targets (this results in equal percentage improvements from one phase to the next). After program start, we plan to measure a new baseline using the data to be supplied by the Government. Based on that baseline, we will work with the Program Manager to develop new interim performance targets.

	% P(Miss)	% P(FA)	% TE
<b>SAD Base</b>	5.0	6.0	5.5
<b>Phase 1</b>	4.2	3.3	3.7
<b>Phase 2</b>	3.6	1.8	2.4
<b>Phase 3</b>	3.0	1.0	1.5
<b>LID Base</b>	10.0	15.0	12.0
<b>Phase 1</b>	7.9	6.1	6.9
<b>Phase 2</b>	6.3	2.5	3.5
<b>Phase 3</b>	5.0	1.0	1.7
<b>SID Base</b>	13.0	5.0	7.2
<b>Phase 1</b>	11.9	3.3	5.2
<b>Phase 2</b>	10.9	2.2	3.7
<b>Phase 3</b>	10.0	1.5	2.6
<b>KWS Base</b>	40.0	6.0	10.4
<b>Phase 1</b>	28.8	4.2	7.3
<b>Phase 2</b>	20.8	2.9	5.1
<b>Phase 3</b>	15.0	2.0	3.5

**Table 3-1 Proposed phased performance targets.**  
TE is the harmonic mean of P(Miss) and P(FA).

The BAA specifies that, for SID, a hit is the correct identification of any of the 10 HVIs in a set. While the P(miss) for a set is somewhat less than that for a single HVI, the P(FA) for a set grows almost linearly with the number of target speakers. (A similar statement can be made for KWS where a hit is defined as the correct identification of any word in a desired set.) A detailed analysis of this effect is provided in Appendix B. We propose to work with the Program Manager to clarify the intended measure of performance.

### 3.5 Software Integration and Deliveries

BBN will deliver to the Government two systems. Six weeks prior to the end of each phase, BBN will deliver an *evaluation system* to the Evaluation Team. During each of Phases 2 and 3, BBN will deliver a fully functioning *trainable system* that is military user-friendly, with an intuitive, easy-to-use



graphical user interface (GUI), and that will be field-trainable for new languages, speakers, and keywords.

These delivery requirements present challenges that BBN is uniquely equipped to handle. We are highly experienced in the integration of technology from other sites into functioning systems, including trainable systems, with GUIs, and delivering real-time, end-to-end systems to Government and other customers, either for internal use or for deployment in the field for 24/7 operations. Section 9.4.1 details our experience in the building and delivery of speech-related operational systems, and our plans in this effort to integrate and test software from different sites, including the specification of appropriate APIs for the delivery of software from other sites to BBN, consistent with the delivery requirements of the Evaluation Team. (Also, Section 9.4.2 summarizes BBN's secure processing capabilities, in case they are needed in this project.)

We are fortunate that our team members have had experience delivering their software for external evaluation and integration purposes. For the evaluation system, we plan to integrate the best algorithms from the team members, as they become available. Currently, we do not expect speed to be an issue for running the system in detection-recognition mode. However, should speed become a concern for any of the software components, we will work with our software developers to find suitable speedups. Again, BBN has had extensive experience in speeding up speech-related software to real-time or faster.

For the trainable system, we will work with the Program Manager and potential users to help specify the interface and training data requirements for each of the four RATS applications. Based on extensive testing, we will then decide which components to include in the trainable system, so that we can deliver

the best performing system within the constraints of third-party operation and speed. During Phase 2, we will place in the hands of users an initial trainable system. In order to ensure that the system meets the needs of the users, we will work closely with the users to get their feedback and suggestions for enhancing the system's capabilities and for improving the user interface. Improvements that can be implemented easily will be done immediately and another version will be delivered. Other suggestions, as well as further improvements in the technology, will be incorporated in the system that will be delivered during Phase 3.

### 3.6 Speech Transcription Option

The 'T' in the acronym RATS stands for 'transcription'. Yet, none of the tasks in the RATS program focuses explicitly on the transcription of speech into a sequence of words. Even though the short-term goals of RATS do not include transcription, since it may be viewed as too difficult a task on which high accuracy could be obtained in the short-term, we have found that KWS users prefer to have a full transcription – even if errorful – to avoid having to listen to every possible hit. Although we plan to use elements of modeling for speech transcription to perform KWS, we believe it is important to work on the full transcription task itself, not only for its own sake, but because it will also lead to better KWS performance. As such, we are proposing, as an option for Phases 2 and 3 of this program, to work explicitly on the transcription task, at least for one language. (The language will most likely be Arabic, but it would be decided in consultation with the Program Manager, should this option be exercised.) BBN and Cambridge University, both highly qualified to work on this task, will join efforts in using the various noise amelioration techniques presented in this proposal to perform research on speech transcription in noise.



## 4 Comparison with Current Technology

We review current approaches to noise compensation, SAD, SID, LID, and KWS, and contrast them briefly to the proposed methods (see Section 3). We also provide state-of-the-art performance results under both clean and noisy conditions. Results with communication noise in four languages (Iraqi Arabic, Farsi, Dari, and Pashto) from the DARPA-funded RATS feasibility study by BBN and MIT Lincoln Lab are also given (see baseline numbers in Table 3-1).

### 4.1 Noise Compensation

Two basic approaches to dealing with noise are: enhancing the features (feature-based) and dynamically adjusting model parameters (model-based). One early feature-based approach is spectral subtraction [46][4] which tries to remove an estimate of the noise spectrum from the noisy speech spectrum. Recent extensions attempt to build statistical mappings from the noisy spectrum to the clean spectrum [47,48,20]. Model-based approaches, such as Vector Taylor Series [23-24], Parallel Model Combination [22], and predictive CMLLR [25], adapt the HMM state observation probabilities based on an estimate of the noise. These methods were mostly developed for testing automatic speech recognition (ASR) under mis-matched conditions, using standard MFCC or PLP front-ends. Researchers have also developed long-span features based on human auditory models [7][11] that are generally more robust to noise, without requiring an explicit noise estimate.

While each of the above methods has been shown to provide significant benefits to ASR in noise, there has not been a systematic effort to apply and optimize these techniques in combination. In Section 3.2 we presented a unified framework for attacking noise across all four RATS applications. This included a number of robust feature extraction and noise

compensation techniques, and innovative discriminative training methods for reducing feature dimensionality and for building detailed multi-resolution models of speech, speakers, and languages. In addition, we proposed a novel condition-adaptive strategy for fusing decisions from multiple classifiers, as well as unsupervised confidence-based model adaptation, in order to reduce uncertainty in the presence of noise.

### 4.2 Speech Activity Detection

In the NIST Rich Transcription evaluation, the best SAD systems [49-51], using a combination of energy- and spectral-based features, and classifiers based on Gaussian Mixture Models (GMM) achieved an error rate (% of total time that was misclassified) of 3.2% when tested on the Meeting and Lecture data. Their performance degraded to 10% when tested on other noisy data. In the RATS feasibility study, the error rate for GMM-based SAD was 11% ( $P_{\text{miss}}=5\%$  at  $P_{\text{FA}}=6\%$ ). Recently, in a speech/nonspeech classification (nondetection) task, our partners Shamma and group [19] showed that an SVM classifier trained on multi-D cortical features (Section 3.2.1) achieved an error rate of 1.2% on noisy speech – significantly better than other approaches [52-53].

In addition to using robust features, our proposed methods place greater reliance on detecting speech by virtue of its predictable time varying nature, which can be modeled by the long-term modulation spectrum and also by using longer-duration tokenizations of speech, such as phonemes or self-organized units (SOUs). In Section 3.3.1 we described a novel factorial HMM for SAD that can achieve even greater robustness by modeling the effects of noise explicitly.

### 4.3 Speaker Identification

In SID, broad speaker characteristics are represented by text-independent statistical models such as GMMs [54-55] that can be discriminatively trained [42]; or SVM



classifiers that use speaker adaptation features (such as MLIR matrices [38]) or adapted GMM supervector means [56]. Joint Factor Analysis (JFA) [28][57] and projection approaches, such as Nuisance Attribute Projection (NAP) [58], are used to reduce the effect of inter-session channel variability. Higher level features, such as word usage, pronunciation, and prosodic patterns are used when available [59]. Multiple systems are used with non-linear classifiers, such as logistic regression or neural network, to form a single decision [60-61].

In the 2008 NIST Speaker Recognition Evaluations, the best results in the telephone matched train/test condition, obtained by our partners at Brno University of Technology, were  $P_{\text{miss}}=5\%$  at  $P_{\text{FA}}=5\%$ . In the RATS feasibility study, the baseline result under noise was  $P_{\text{miss}}=13\%$  at  $P_{\text{FA}}=5\%$ .

The main weakness of current SID practices is that the acoustic likelihood ratios are based on simple GMMs, which are very susceptible to noise and cannot model distinguishing long-term speaker attributes. Our proposed approach overcomes this problem by computing acoustic likelihood ratios based on all levels of tokenization, using detailed HMMs trained via novel discriminative training, as outlined in Section 3.3.2.

#### 4.4 Language Identification

LID is commonly performed using techniques similar to SID – such as discriminatively trained GMMs and supervector SVMs – in addition to phonotactic approaches that use n-gram scores from a phoneme recognizer [41][62]. As in SID, these systems are combined to optimize performance. In the 2009 NIST Language Recognition Evaluation, the best results were  $P_{\text{miss}}=1\%$  at  $P_{\text{FA}}=1\%$  for 30 second tests. In the RATS feasibility study, the baseline results under noise were  $P_{\text{miss}}=10\%$  at  $P_{\text{FA}}=15\%$ .

Our proposal for LID (Section 3.3.3) replaces weak GMMs with powerful multi-resolution

HMM tokenizers that are trained discriminatively and in a speaker-adaptive manner in order to model and factor out inter-speaker variability. The use of semi-supervised training and SOUs will enable us to produce tokenizers for all 15 of the languages in the RATS program. We also propose more refined phonotactic models that look beyond local n-gram context for broad phonotactic patterns that are characteristic of languages.

#### 4.5 Keyword Spotting

KWS is commonly performed with either phoneme or word recognition. In [63], we demonstrated that modeling context (words next to the keywords) significantly improved performance. In [35], we showed that using lattices in KWS with appropriately set detection thresholds is also important.

In the 2006 NIST Spoken Term Detection evaluation, for the clean conversational telephone speech domain in which all sites participated, the best performing system, submitted by BBN [35] using large vocabulary recognition, achieved  $P_{\text{miss}}=10\%$  at  $P_{\text{FA}}=0.01\%$  [64]. In the RATS feasibility study, the Arabic KWS system was trained with 1 hour of transcribed noisy data. Using other out-of-domain training data and applying adaptation techniques, the KWS error rates for Iraqi Arabic were  $P_{\text{miss}}=40\%$  at  $P_{\text{FA}}=6\%$ .

Our proposed work (Section 3.3.4) includes stronger prior models, such as topic-weighted language models, and robustness through the use of multi-resolution models, with condition-dependent fusion and model adaptation. We proposed several techniques, based on unsupervised adaptation, for improving both the acoustic and language models for keywords that are not well-represented in the training data. Semi-supervised training would allow us to dramatically improve performance when we have very little transcribed noisy audio.



## 5 Statement of Work

BBN proposes to build a system, PATROL, that will revolutionize US ability to pinpoint speech activity, identify the language and the speaker, and detect keywords of interest, under very noisy and degraded conditions. The work presented in Section 3 will be accomplished in three phases: an 18-month Phase 1, a 12-month Phase 2, and a 12-month Phase 3.

Deliverables under this effort shall include:

- Quarterly Progress Reports (QPR)
- Final report
- Papers published in international conferences and/or journals during the course of the program
- All software and documentation developed under the project
- Any data collected under the project
- Integrated systems for evaluation (6 weeks before end of each phase)
- Trainable, military user-friendly system during Phases 2 and 3.

In the remainder of this section we present an itemization of the proposed tasks and subtasks, organized by phase, and grouped into one of the following areas: software integration and delivery, feature extraction and projection, noise compensation, speech activity detection (SAD), language identification (LID), speaker identification (SID), and keyword spotting (KWS). In Phases 2 and 3 we also list tasks associated with the proposed RATS Speech Transcription option described in Section 3.6.

For each subtask, we show its duration (*Dur*), objective (*Obj*), the approach taken to accomplish it (*App*), participating sites (*Sites*), completion criteria (*Criteria*), and deliverables (*Del*). We also indicate any dependencies (*Dep*) to other tasks. All acronyms are as defined in Section 3.

### 1. Phase 1

#### 1.1. Software Integration and Delivery

##### 1.1.1. Develop library for FDLP and modulation spectrum

*Dur*: 2 months. *Obj*: Develop portable software library for computing FDLP and modulation spectrum. *App*: Develop C++ library that computes FDLP and modulation spectrum; write associated regression tests; optimize code for speed. *Sites*: JHU. *Criteria*: Code compiling and passing regression tests at all sites. *Del*: QPR.

##### 1.1.2. Develop library for testing MLPs

*Dur*: 3 months. *Obj*: Develop software library that provided functions for testing multi-layer perceptrons (MLPs). *App*: Develop C++ library functions that can test MLPs on a corpus of feature vectors; write associated regression tests; optimize code for speed. *Sites*: JHU. *Criteria*: Library compiles and passes regression tests at all sites. *Del*: QPR. *Dep*: 1.1.1.

##### 1.1.3. Develop library for cortical feature extraction

*Dur*: 2 months. *Obj*: Develop portable software library for computing cortical features. *App*: Develop C++ library that computes cortical features; write associated regression tests; optimize code for speed. *Sites*: UMD. *Criteria*: Code compiling and passing regression tests at all sites. *Del*: QPR.

##### 1.1.4. Integrate baseline UMD SAD SVM to PATROL

*Dur*: 2 months. *Obj*: Integrate SAD SVM training and testing software to BBN's system. *App*: Develop C++ library functions that can train and test SAD SVMs on a corpus of feature vectors with associated targets; check source code into BBN's software repository; write associated regression tests; optimize code for speed. *Sites*: UMD. *Criteria*: Code working properly at BBN. *Del*: QPR. *Dep*: 1.1.3.



### 1.1.5. Transition baseline SID and LID systems from BUT to BBN

**Dur:** 6 months. **Obj:** Integrate BUT SID and LID system components to BBN's system.

**App:** Develop C++ library functions that can test BUT's SID and LID models on existing LDC/NIST corpora; check source code into BBN's software repository; write associated regression tests. **Sites:** BUT. **Criteria:** Obtain same results as BUT on NIST SID/LID evaluation sets. **Del:** QPR.

### 1.1.6. Integrate PATROL research components into Phase 1 Deliverable system

**Dur:** 7 months. **Obj:** Continuously integrate research components into PATROL system. Ensure that PATROL system runs within specified time limits in evaluation mode, on deliverable hardware. **App:** Iteratively integrate system components to PATROL system as they become updated by research efforts. Test component connections and measure performance characteristics (accuracy, speed). Optimize components as needed in order to meet target speed. **Sites:** BBN. **Criteria:** PATROL system achieves expected accuracy and runs within specified time constraints. Task repeats until the start of Phase 1 evaluation. **Del:** QPR.

### 1.1.7. Develop GUI for evaluation system

**Dur:** 5 months. **Obj:** Develop a graphical user interface (GUI) for the PATROL system, supporting Phase 1 evaluation tasks. **App:** Implement GUI that allows the user to easily select an evaluation task, open a test file, run the PATROL system on it, and save the output. Write source code that interprets GUI inputs and calls corresponding functions in the PATROL system library. **Sites:** BBN. **Criteria:** GUI is intuitive, easy to use and allows user to perform all evaluation tasks. **Del:** QPR.

### 1.1.8. Test and prepare P1 PATROL system for Delivery

**Dur:** 2 months. **Obj:** Ensure that Phase 1 PATROL system runs with expected accuracy

and speed in evaluation mode. **App:** Install PATROL system on hardware to be delivered to Evaluation Team. Run PATROL system in evaluation mode on all tasks. Deliver system to Evaluation Team. **Sites:** BBN. **Criteria:** PATROL system runs as expected on deliverable hardware. **Del:** QPR; software and hardware delivery to Evaluation Team on 1/2/2012. **Dep:** 1.1.6, 1.1.7.

## 1.2. Feature Extraction and Projection

### 1.2.1. Project high-dimensional features via MLP

**Dur:** 10 months. **Obj:** Determine whether cortical and modulation spectrum features work well with MLPs. Also determine whether MLPs can scale to high-dimensional features. **App:** Feed high dimensional cortical and modulation spectrum features into MLP and train MLP for each of the four RATS applications; explore different MLP architectures and training methods; optimize code for speed; support MLP integration into PATROL system. **Sites:** JHU. **Criteria:** MLP-derived features perform better than PCA/LDA-projected features for each of the four RATS applications. **Del:** QPR. **Dep:** 1.1.2.

### 1.2.2. Explore feature-based fusion

**Dur:** 8 months. **Obj:** Determine optimal procedure for combining/fusing different types of features (e.g., PLP, FDLP, MLP). **App:** Investigate two methods for fusing multiple feature streams: (a) feature concatenation and MLP-based projection, and (b) Dempster-Shafer fusion. Run experiments on RATS data to determine which method works best for each of the four RATS applications. Develop software library that supports feature fusion functions. Optimize code for speed. **Sites:** JHU. **Criteria:** Improved results across all four RATS applications from fusing multiple feature streams. **Del:** QPR.



### 1.3. Noise Compensation

#### 1.3.1. Enhance cortical features

**Dur:** 13 months. **Obj:** Develop methods for noise suppression in the cortical domain. **App:** Implement filter that suppresses regions of noise in the cortical domain. Estimate parameters of filter on RATS data. Investigate filters that adapt their parameters based on an estimate of the noise. Periodically integrate and update enhancement filters into cortical feature extraction library. Coordinate with JHU on projecting enhanced cortical features via MLPs. Optimize code for speed. **Sites:** UMD. **Criteria:** Improved performance of RATS applications compared to using un-enhanced cortical features. **Del:** QPR.

#### 1.3.2. Investigate VTS for noise adaptation

**Dur:** 13 months. **Obj:** Investigate vector Taylor series (VTS) for adapting HMMs to noise on RATS data. **App:** Consult with Cambridge University on implementation of VTS noise adaptation. Incorporate VTS in models used across all four RATS applications. Measure performance with and without VTS. Repeat measurements after incorporating MLP-projected enhanced features from UMD/JHU. Optimize implementation for speed. **Sites:** BBN, CU. **Criteria:** Improved performance of RATS applications compared to only using feature-based noise compensation. **Del:** QPR. **Dep:** 1.1.2.

### 1.4. SAD

#### 1.4.1. Develop cortical domain SVM for SAD

**Dur:** 12 months. **Obj:** Develop cortical domain SAD system based on Support Vector Machines (SVMs). **App:** Investigate the use of SVMs for speech/non-speech audio classification. Run experiments on RATS data as it becomes available. Initially, make use of cortical features with PCA/LDA dimensionality reduction. Later, use MLP-based projection from JHU. Release updated PATROL software libraries for SVM testing. **Sites:** UMD. **Criteria:** Improved SAD

performance on RATS data compared to standard speech/non-speech GMM baseline. **Del:** QPR. **Dep:** 1.1.4.

#### 1.4.2. Build SOU HMM for SAD

**Dur:** 9 months. **Obj:** Develop SAD system based on self-organized unit (SOU) hidden Markov models (HMMs). **App:** Apply the process of clustering acoustic trajectories in RATS data into a number of SOUs for speech and non-speech. Design and train an HMM for each labeled SOU in the RATS training data. Contrast such system with a baseline speech/non-speech GMM SAD. Integrate MLP-projected features from UMD/JHU as they become available. Optimize SOU HMM SAD system for speed. **Sites:** BBN. **Criteria:** Improved SAD performance on RATS data compared to simple speech/non-speech GMM baseline. **Del:** QPR. **Dep:** 1.1.2.

### 1.5. LID

#### 1.5.1. Establish baseline BUT LID system on RATS data

**Dur:** 5 months. **Obj:** Establish a baseline BUT LID system tuned on RATS data. **App:** Train and test all components of BUT LID system on RATS data. Optimize each component for speed and accuracy. Iteratively integrate optimized components to PATROL system. **Sites:** BUT. **Criteria:** BUT LID system runs properly on RATS data and achieves improved performance compared to initial un-tuned LID system. **Del:** QPR.

#### 1.5.2. Incorporate MLP features in BUT LID system

**Dur:** 6 months. **Obj:** Test MLP-projected cortical or modulation spectrum features in LID system. **App:** Periodically integrate UMD/JHU's MLP-projected cortical or modulation spectrum features into BUT LID system and re-optimize LID on RATS data. Compare against current best baseline on RATS development set. **Sites:** BUT. **Criteria:** Perform periodically until best front-end processing is integrated to Phase 1 BUT LID system components. **Del:** QPR. **Dep:** 1.1.2, 1.5.1.



### 1.5.3. Develop SOU-based LID system

**Dur:** 13 months. **Obj:** Develop SOU-based BBN LID system. **App:** Tokenize RATS training data based on self-organized units (SOU). Estimate language-specific SOU GMM and HMM acoustic models. Train language-specific SOU n-gram models. Optimize LID system components for speed and accuracy. Incorporate MLP features as they become available. **Sites:** BBN. **Criteria:** BBN LID system runs properly on RATS data and achieves performance comparable to or better than BUT's LID system on RATS devset. **Del:** QPR. **Dep:** 1.1.2.

## 1.6. SID

### 1.6.1. Develop JHU SID system

**Dur:** 12 months. **Obj:** Develop JHU SID system tuned on RATS data. **App:** Train and test all components of JHU SID system on RATS data. Optimize each component for speed and accuracy. Incorporate MLP features as they become available. Iteratively integrate optimized components to PATROL system. **Sites:** JHU. **Criteria:** JHU SID system runs properly on RATS data and achieves improved performance compared to initial un-tuned SID system. **Del:** QPR. **Dep:** 1.1.2.

### 1.6.2. Establish baseline BUT SID system on RATS data

**Dur:** 5 months. **Obj:** Establish a baseline BUT SID system tuned on RATS data. **App:** Train and test all components of BUT SID system on RATS data. Optimize each component for speed and accuracy. Iteratively integrate optimized components to PATROL system. **Sites:** BUT. **Criteria:** BUT SID system runs properly on RATS data and achieves improved performance compared to initial un-tuned SID system. **Del:** QPR.

### 1.6.3. Incorporate MLP features in BUT SID system

**Dur:** 6 months. **Obj:** Test MLP-projected cortical or modulation spectrum features in LID system. **App:** Periodically integrate UMD/JHU's MLP-projected cortical or

modulation spectrum features into BUT LID system and re-optimize LID on RATS data. Compare against current best baseline on RATS development set. **Sites:** BUT. **Criteria:** Perform periodically until best front-end processing is integrated to Phase I BUT LID system components. **Del:** QPR. **Dep:** 1.1.2, 1.6.2.

### 1.6.4. Try BBN's ASR model adaptation parameters as features for SID

**Dur:** 4 months. **Obj:** Investigate the use of BBN's ASR speaker adaptation parameters as features in SID. **App:** Train SID classifiers that rely on ASR speaker adaptation parameters (MLLR, CMLLR matrices) provided by BBN. Test on RATS development data. Repeat investigation periodically as BBN's ASR system evolves during Phase I. **Sites:** BUT. **Criteria:** Perform periodically until best SID classification results are obtained in time for integration to final Phase I PATROL system. **Del:** QPR.

### 1.6.5. Develop SOU-based SID system

**Dur:** 13 months. **Obj:** Develop SOU-based BBN SID system. **App:** Tokenize RATS training data based on self-organized units (SOU). Estimate universal background SOU HMM (UBM). Train speaker-dependent SOU HMM acoustic models. Train speaker-specific SOU n-gram models. Optimize SID system components for speed and accuracy. Incorporate MLP features as they become available. **Sites:** BBN. **Criteria:** BBN SID system runs properly on RATS data and achieves performance comparable to or better than BUT's SID system on RATS devset. **Del:** QPR.

## 1.7. KWS

### 1.7.1. Establish baseline KWS using LVCSR

**Dur:** 9 months. **Obj:** Develop Arabic and Farsi KWS systems using large vocabulary recognizers (LVCSR). **App:** Demonstrate that the performance is better than using phonemes or sub-words, and better than using a small vocabulary that includes only the



keywords plus sub-words. Initially, run experiments on existing Arabic LDC corpora. When RATS data becomes available, retrain KWS system and measure performance on RATS Arabic and Farsi devsets. *Sites:* BBN, CU. *Criteria:* A KWS system using LVCSR. Comparative results with phoneme based KWS. *Del:* QPR.

#### 1.7.2. Integrate MLP features from UMD/JHU

*Dur:* 6 months. *Obj:* Integrate improved MLP-projected features from UMD and JHU. *App:* Periodically evaluate the impact of UMD/JHU MLP-projected features on KWS. After each feature update, retrain and test KWS systems on RATS data. Optimize system components as needed on the new features. *Sites:* BBN, CU. *Criteria:* Improved KWS performance due to the use of more advanced front-ends. *Del:* QPR. *Dep:* 1.1.2, 1.7.1.

#### 1.7.3. Develop confidence measures for keywords

*Dur:* 5 months. *Obj:* Develop good confidence measures for keywords on noisy data. *App:* Associate each keyword with features extracted from phonetic lattices. Train classifier (e.g., generalized linear model) on RATS annotated data to predict probability of the recognized keyword being correct. *Sites:* BBN, CU. *Criteria:* Obtain confidence measures that can be used to decide whether to return a recognized keyword. *Del:* QPR. *Dep:* 1.7.1.

#### 1.7.4. Develop adaptation procedures for KWS

*Dur:* 6 months. *Obj:* Investigate adaptation of KWS tokenizers and classifiers for improved accuracy. *App:* Use high-confidence detected keywords on test set in order to adapt KWS models. Re-decode test data with adapted models. Optimize adaptation and recognition stages for speed. *Sites:* BBN. *Criteria:* Improved KWS performance due to confidence-based adaptation. *Del:* QPR. *Dep:* 1.7.3.

#### 1.7.5. Generate pronunciations for OOV keywords

*Dur:* 4 months. *Obj:* Generate set of alternative pronunciations for keywords that are not in the training lexicon. *App:* Explore two approaches to handling unknown keywords. First, use graphemic ASR models, that map each letter in a word to a phoneme. Second, develop algorithms for generating a set of plausible pronunciations from the surface word form, and use these pronunciations within phonetic ASR models for KWS. *Sites:* BBN. *Criteria:* Ability to spot words that are outside of the training lexicon. *Del:* QPR. *Dep:* 1.7.1.

### 2. Phase 2

#### 2.1. Software Integration and Delivery

##### 2.1.1. Develop system for training in the field

*Dur:* 10 months. *Obj:* Develop infrastructure to support training in the field for all four RATS applications. *App:* Start with snapshot of Phase 1 PATROL system; integrate all BBN training modules into deliverable pipeline; provide GUI-accessible functions that perform training on given data; test training pipeline extensively on deliverable hardware; optimize components for speed; write supporting documentation. *Sites:* BBN. *Criteria:* Deliverable system performs all training tasks reliably and efficiently. *Del:* QPR; software and hardware delivery to Government on 1/2/2013.

##### 2.1.2. Develop GUI for trainable system

*Dur:* 5 months. *Obj:* Develop a graphical user interface (GUI) for the trainable version of the PATROL system. *App:* Implement military user friendly GUI that allows the user to train and test the PATROL system on a given set of data. Write source code that interprets GUI inputs and calls corresponding functions in the PATROL system library. *Sites:* BBN. *Criteria:* GUI is intuitive, easy to use and allows user to perform all training/testing tasks. *Del:* QPR; software. *Dep:* 2.1.1.



### 2.1.3. Integrate PATROL research components into Phase 2 deliverable system

**Dur:** 7 months. **Obj:** Continuously integrate research components into PATROL system. Ensure that PATROL system runs within specified time limits in training/evaluation mode, on deliverable hardware. **App:** Iteratively integrate system components to PATROL system as they become updated by research efforts. Test component connections and measure performance characteristics (accuracy, speed). Optimize components as needed in order to meet target speed. **Sites:** BBN. **Criteria:** PATROL system achieves expected accuracy and runs within specified time constraints. Task repeats until the start of Phase 2 evaluation. **Del:** QPR.

### 2.1.4. Test and prepare P2 PATROL system for delivery

**Dur:** 2 months. **Obj:** Ensure that Phase 2 PATROL system runs with expected accuracy and speed in training/evaluation mode. **App:** Install PATROL system on hardware to be delivered to Government and Evaluation Team. Run PATROL system in training/evaluation mode on all tasks. Deliver system to Government and Evaluation Team. **Sites:** BBN. **Criteria:** PATROL system runs as expected on deliverable hardware. **Del:** QPR; software and hardware delivery to Government and Evaluation Team on 1/2/2013. **Dep:** 2.1.1, 2.1.2, 2.1.3.

## 2.2. Feature Extraction and Projection

### 2.2.1. Explore hierarchical MLP training

**Dur:** 7 months. **Obj:** Determine whether MLP-based projection can be improved via hierarchical training. **App:** Use output of trained MLP as input to second level MLP; optimize code for speed; support MLP integration into PATROL system. **Sites:** JHU. **Criteria:** Improved classification performance in RATS applications. **Del:** QPR.

### 2.2.2. Project high-dimensional features via RDT

**Dur:** 12 months. **Obj:** Determine whether RDT-based feature projection is better than MLPs. **App:** Train discriminative RDT to project UMD/JHU features to lower dimensional sub-space for each of the four RATS applications; optimize code for speed. **Sites:** BBN. **Criteria:** Improved classification performance in RATS applications. **Del:** QPR.

## 2.3. Noise Compensation

### 2.3.1. Investigate RDT-based enhancement

**Dur:** 12 months. **Obj:** Apply RDT to cortical spectrum in order to suppress the noise. **App:** Estimate clean speech model from clean corpus; Initialize RDT mapping based on UMD's noise filter; Train RDT discriminatively on RATS data to improve classification with clean speech model. **Sites:** BBN,UMD. **Criteria:** Improved classification performance in RATS applications. **Del:** QPR.

### 2.3.2. Investigate JUD and PCMLLR for noise adaptation

**Dur:** 10 months. **Obj:** Investigate joint uncertainty decoding (JUD) and predictive CMLLR (PCMLLR) for adapting HMMs to noise on RATS data. **App:** Consult with Cambridge University on implementation of JUD and PCMLLR noise adaptation. Incorporate compensated models used across all four RATS applications. Contrast performance with VTS noise compensation. Repeat measurements after incorporating MLP/RDT-projected enhanced features from UMD/JHU. Optimize implementation for speed. **Sites:** BBN,CU. **Criteria:** Improved performance of RATS applications compared to only using feature-based noise compensation. **Del:** QPR.

### 2.3.3. Investigate condition-adaptive fusion

**Dur:** 8 months. **Obj:** Determine optimal procedure for combining/fusing scores and features from multiple classifiers (e.g., HMMs, GMMs). **App:** Use measured features of input (e.g., SNR) as additional features in final classification stage, along



with scores from other classifiers. Develop software library that supports score fusion functions. Contrast with Dempster-Shafer methods. Optimize code for speed. *Sites:* JHU. *Criteria:* Improved results across all four RATS applications from fusing multiple classifier scores. *Del:* QPR.

## 2.4. SAD

### 2.4.1. Build phoneme-based HMM for SAD

*Dur:* 6 months. *Obj:* Develop SAD system based on phoneme-class HMMs. *App:* Use semi-supervised training methods to label RATS SAD training data at phoneme level. Design and train an HMM for each class of phonemes. Contrast such system with the SOU HMM SAD. Integrate MLP/RDT-projected features from UMD/JHU as they become available. Optimize HMM SAD system for speed. *Sites:* BBN. *Criteria:* Improved SAD performance on RATS data compared to simple speech/non-speech GMM baseline. *Del:* QPR.

### 2.4.2. Investigate adaptation for SAD

*Dur:* 5 months. *Obj:* Improve SAD performance through unsupervised adaptation. *App:* Use high-confidence speech/non-speech boundaries detected on test data to adapt acoustic models and then retest with adapted models. *Sites:* BBN. *Criteria:* Improved SAD performance on RATS data compared to unadapted SAD baseline. *Del:* QPR.

## 2.5. LID

### 2.5.1. Develop phoneme-based LID system

*Dur:* 12 months. *Obj:* Develop phoneme-based BBN LID system. *App:* Use semi-supervised training methods to label RATS training data based on phonemes. Estimate language-specific phoneme HMM acoustic models. Train language-specific phoneme n-gram models. Optimize LID system components for speed and accuracy. Incorporate MLP/RDT-projected features as they become available. *Sites:* BBN. *Criteria:* Comparable performance to SOU-based LID. *Del:* QPR.

### 2.5.2. Discriminative training of HMMs for LID

*Dur:* 12 months. *Obj:* Explore discriminative training of HMMs for LID. *App:* Estimate HMM parameters so as to maximize the posterior probability of the correct language on RATS training data. Incorporate MLP/RDT-projected features as they become available. *Sites:* BBN. *Criteria:* Improved performance compared to maximum likelihood HMM LID. *Del:* QPR.

### 2.5.3. Improve JFA in LID

*Dur:* 12 months. *Obj:* Investigate the use of joint factor analysis (JFA) for modeling noise variability in LID. *App:* Apply JFA on RATS data. Extend JFA framework to account for noise variability. *Sites:* BUT. *Criteria:* Improved performance compared to baseline BUT system. *Del:* QPR.

## 2.6. SID

### 2.6.1. Develop phoneme-based SID system

*Dur:* 12 months. *Obj:* Develop phoneme-based BBN SID system. *App:* Tokenize RATS training data based on phonemes. Estimate universal background phoneme HMM (UBM). Train speaker-dependent phoneme HMM acoustic models. Train speaker-specific phoneme n-gram models. Optimize SID system components for speed and accuracy. Incorporate MLP features as they become available. *Sites:* BBN. *Criteria:* BBN SID system runs properly on RATS data and achieves performance comparable to or better than BUT's SID system on RATS devset. *Del:* QPR.

### 2.6.2. Discriminative training of HMMs for SID

*Dur:* 9 months. *Obj:* Explore discriminative training of HMMs for SID. *App:* Estimate HMM parameters so as to maximize the posterior probability of the correct speaker on RATS training data. Incorporate MLP/RDT-projected features as they become available. *Sites:* BBN, JHU. *Criteria:* Improved performance compared to maximum likelihood HMM SID. *Del:* QPR.



### 2.6.3. Improve JFA in SID

**Dur:** 12 months. **Obj:** Investigate the use of joint factor analysis (JFA) for modeling noise variability in LID. **App:** Apply JFA on RATS data. Extend JFA framework to account for noise variability. **Sites:** BUT. **Criteria:** Improved performance compared to baseline BUT system. **Del:** QPR.

### 2.7. KWS

#### 2.7.1. Investigate semi-supervised training methods

**Dur:** 10 months. **Obj:** Explore semi-supervised training methods for KWS acoustic model training. **App:** Build initial acoustic models on transcribed RATS corpus, then decode large un-transcribed audio and train on select automatic transcripts. **Sites:** BBN. **Criteria:** Improved KWS performance compared to training on manually transcribed data. **Del:** QPR.

#### 2.7.2. Use topic-weighted LM

**Dur:** 8 months. **Obj:** Investigate topic-weighted language model (LM) for KWS. **App:** Run UTD on transcribed text for the language to derive a set of topics. Run recognition on test passage; detect topic automatically; derive topic-weighted LM; rerun recognition. **Sites:** BBN. **Criteria:** Improved KWS performance compared to using a generic LM. **Del:** QPR.

### 2.8. Speech Transcription Option

#### 2.8.1. Establish ASR baseline on RATS devset

**Dur:** 5 months. **Obj:** Establish training/test sets and proper ASR baseline using RATS data. **App:** Coordinate with Data Team to form a proper RATS ASR development set; build initial RATS ASR system on designated training corpus. **Sites:** BBN, CU. **Criteria:** Reasonable ASR accuracy on RATS data. **Del:** QPR.

#### 2.8.2. Develop ASR RATS system

**Dur:** 7 months. **Obj:** Further improve accuracy of PATROL ASR system. **App:** Incorporate features and noise compensation methods developed for KWS as they become

available. Explore discriminative training and adaptation methods. **Sites:** BBN, CU. **Criteria:** Improved ASR accuracy compared to baseline ASR system on RATS data. **Del:** QPR. **Dep:** 2.8.1.

### 3. Phase 3

#### 3.1. Software Integration and Delivery

##### 3.1.1. Develop system for training in the field

**Dur:** 10 months. **Obj:** Develop infrastructure to support training in the field for all four RATS applications. **App:** Integrate training modules from UMD/JHU/BUT into deliverable pipeline; test training pipeline extensively on deliverable hardware; optimize components for speed; write supporting documentation. **Sites:** BBN. **Criteria:** Deliverable system performs all training tasks reliably and efficiently. **Del:** QPR; software and hardware.

##### 3.1.2. Integrate PATROL research components into Phase 3 deliverable system

**Dur:** 10 months. **Obj:** Continuously integrate research components into PATROL system. Ensure that PATROL system runs within specified time limits in training/evaluation mode, on deliverable hardware. **App:** Iteratively integrate system components to PATROL system as they become updated by research efforts. Test component connections and measure performance characteristics (accuracy, speed). Optimize components as needed in order to meet target speed. **Sites:** BBN. **Criteria:** PATROL system achieves expected accuracy and runs within specified time constraints. Task repeats until the start of Phase 3 evaluation. **Del:** QPR. **Dep:** 3.1.1.

##### 3.1.3. Test and prepare P3 PATROL system for Delivery

**Dur:** 2 months. **Obj:** Ensure that Phase 3 PATROL system runs with expected accuracy and speed in training/evaluation mode. **App:** Install PATROL system on hardware to be delivered to Government and Evaluation



Team. Run PATROL system in evaluation mode on all tasks. Deliver system to Government and Evaluation Team. *Sites:* BBN. *Criteria:* PATROL system runs as expected on deliverable hardware. *Del:* QPR. Software and hardware delivery to Government and to Evaluation Team on 1/2/2014. *Dep:* 3.1.1, 3.1.2.

### 3.2. Feature Extraction and Projection

#### 3.2.1. Improve feature extraction methods

*Dur:* 12 months. *Obj:* Improve existing front-end representations. *App:* Analyze results obtained in all four RATS applications and adjust existing or develop new feature extraction modules in order to achieve further improvements to noise robustness; update software libraries; optimize code for speed. *Sites:* UMD, JHU. *Criteria:* Improved robustness to noise. *Del:* QPR.

### 3.3. Noise Compensation

#### 3.3.1. Investigate condition-adaptive fusion

*Dur:* 8 months. *Obj:* Determine optimal procedure for combining/fusing scores and features from multiple classifiers (e.g., HMMs, GMMs). *App:* Continue development of condition-adaptive fusion, investigating new features such as prosody, articulation, phoneme rate. *Sites:* JHU. *Criteria:* Improved results across all four RATS applications from fusing multiple classifier scores. *Del:* QPR.

#### 3.3.2. Investigate RDT-based enhancement

*Dur:* 12 months. *Obj:* Apply RDT to cortical spectrum in order to suppress the noise. *App:* Continue developing RDT based enhancement by investigating different methods for partitioning the cortical feature space into regions. *Sites:* BBN, UMD. *Criteria:* Improved classification performance in RATS applications. *Del:* QPR.

#### 3.3.3. Investigate Noisy CMLLR for noise adaptation

*Dur:* 10 months. *Obj:* Investigate Noisy CMLLR (NCMLLR) for adapting HMMs to noise on RATS data. *App:* Consult with Cambridge University on implementation of

NCMLLR adaptation. Incorporate compensated models used across all four RATS applications. Contrast performance with Phase 2 noise compensation methods. Repeat measurements after incorporating MLP/RDT-projected enhanced features from UMD/JHU. Optimize implementation for speed. *Sites:* BBN, CU. *Criteria:* Improved performance of RATS applications compared to only using feature-based noise compensation. *Del:* QPR.

#### 3.3.4. Develop noise adaptive training methods

*Dur:* 9 months. *Obj:* Derive acoustic models that adapt better to noise. *App:* Apply noisy CMLLR adaptation methods to training in order to obtain canonical, noise-neutral acoustic models; also investigate discriminative methods for estimating the adaptation parameters. *Sites:* BBN, CU. *Criteria:* Improved adaptation performance on test. *Del:* QPR.

### 3.4. SAD

#### 3.4.1. Investigate factorial HMM for SAD

*Dur:* 12 months. *Obj:* Explore the use of factorial HMMs for SAD. *App:* Estimate clean speech phoneme-class models from clean speech corpus; initialize factorial HMM noise states based on identity transforms; train factorial model on RATS data, allowing noise transforms to model the effect of noise on clean speech; investigate both maximum likelihood and discriminative training criteria. *Sites:* BBN. *Criteria:* Improved SAD performance compared to Phase 2 SAD models. *Del:* QPR.

#### 3.4.2. Explore sequence scores for SAD

*Dur:* 8 months. *Obj:* Investigate the use of higher-level features in SAD. *App:* Apply phonotactic-type modeling to learn patterns about the sequences of phonemes or SOUs that are characteristic of speech; fuse such scores with acoustic HMM SAD scores. *Sites:* BBN. *Criteria:* Improved SAD performance compared to Phase 2 SAD models. *Del:* QPR.



### 3.5. LID

#### 3.5.1. Use SAT for LID

**Dur:** 10 months. **Obj:** Use speaker adaptive training (SAT) in LID. **App:** Apply speaker change detection and clustering methods on RATS LID training data; Estimate adaptation transforms for each speaker cluster; incorporate transforms during LID HMM training in order to obtain speaker-neutral models; investigate discriminative training extensions. **Sites:** BBN. **Criteria:** Improved performance compared to standard discriminative HMM LID training. **Del:** QPR.

#### 3.5.2. Use information retrieval methods for LID

**Dur:** 10 months. **Obj:** Explore techniques for global phonotactic modeling in LID. **App:** Collect statistics about phoneme or SOU co-occurrences in speech tokenization output; build models that look beyond local n-gram context; include scores from such models in LID fusion. **Sites:** BUT. **Criteria:** Improved performance compared to baseline LID system. **Del:** QPR.

#### 3.5.3. JFA for HMM-based LID

**Dur:** 12 months. **Obj:** Extend JFA to work with HMM-based SID systems. **App:** Estimate basis of MLLR transformations to model variability in HMM Gaussian parameters; adapt HMMs using JFA-inspired methods. **Sites:** BUT, BBN. **Criteria:** Improved performance compared to baseline HMM SID system. **Del:** QPR.

### 3.6. SID

#### 3.6.1. JFA for HMM-based SID

**Dur:** 12 months. **Obj:** Extend JFA to work with HMM-based SID systems. **App:** Estimate basis of MLLR transformations to model variability in HMM Gaussian parameters; adapt HMMs using JFA-inspired methods. **Sites:** BUT, BBN. **Criteria:** Improved performance compared to baseline HMM SID system. **Del:** QPR.

### 3.7. KWS

#### 3.7.1. Apply SVM phonetic rescoring

**Dur:** 10 months. **Obj:** Use phoneme or word-based SVM models to complement HMMs. **App:** Estimate an SVM for each context-dependent phoneme or word in training; rescore lattice of tokens in test using SVMs; include SVM scores in KWS fusion. **Sites:** BBN. **Criteria:** Improved performance compared to baseline HMM KWS system. **Del:** QPR.

#### 3.7.2. Adapt LM to new keywords

**Dur:** 10 months. **Obj:** Improve estimate of n-gram LM probability for new keywords. **App:** Run KWS system on test data; find high-confidence detections of the new keyword; look at n-gram context; find words that have been observed in similar contexts in LM training; re-estimate LM probability of new keyword based on these similar words; re-run recognition on test, using adapted LM. **Sites:** BBN. **Criteria:** Improved performance compared to baseline generic LM KWS system. **Del:** QPR.

### 3.8. Speech Transcription Option

#### 3.8.1. Develop ASR RATS system

**Dur:** 12 months. **Obj:** Further improve accuracy of PATROL ASR system. **App:** Incorporate features and noise compensation methods developed for KWS as they become available. Explore semi-supervised training methods. **Sites:** BBN, CU. **Criteria:** Improved ASR accuracy compared to baseline ASR system on RATS data. **Del:** QPR.



## 6 Intellectual Property

Any data that BBN collects or acquires under this effort will be made available to all sites in the RATS program.

### Noncommercial Items (Technical Data and Computer Software)

NONCOMMERCIAL ITEMS			
Technical Data and Computer Software To be Furnished With Restrictions	Basis for Assertion	Asserted Rights Category	Name of Person Asserting Restrictions
Software Suite to be developed under the RATS Program including existing libraries as needed	Developed with mixed funding	Negotiated Government Purpose Rights	Raytheon BBN Technologies Corp.

### Commercial Items (Technical Data and Computer Software)

COMMERCIAL ITEMS			
Technical Data and Computer Software To be Furnished With Restrictions	Basis for Assertion	Asserted Rights Category	Name of Person Asserting Restrictions
Byblos Software*	Multiple commercial sales	Commercial	Raytheon BBN Technologies Corp.

\*Byblos Software will be made available to the Government for the duration of the program effort pursuant to a no-cost license.

### Patents

BBN hereby represents that it is the sole or joint owner of the inventions that will be utilized in the proposed program. BBN asserts that it has all of the necessary rights to utilize them to execute the program. BBN will obtain for the Government any necessary licenses in its subcontractor's inventions.

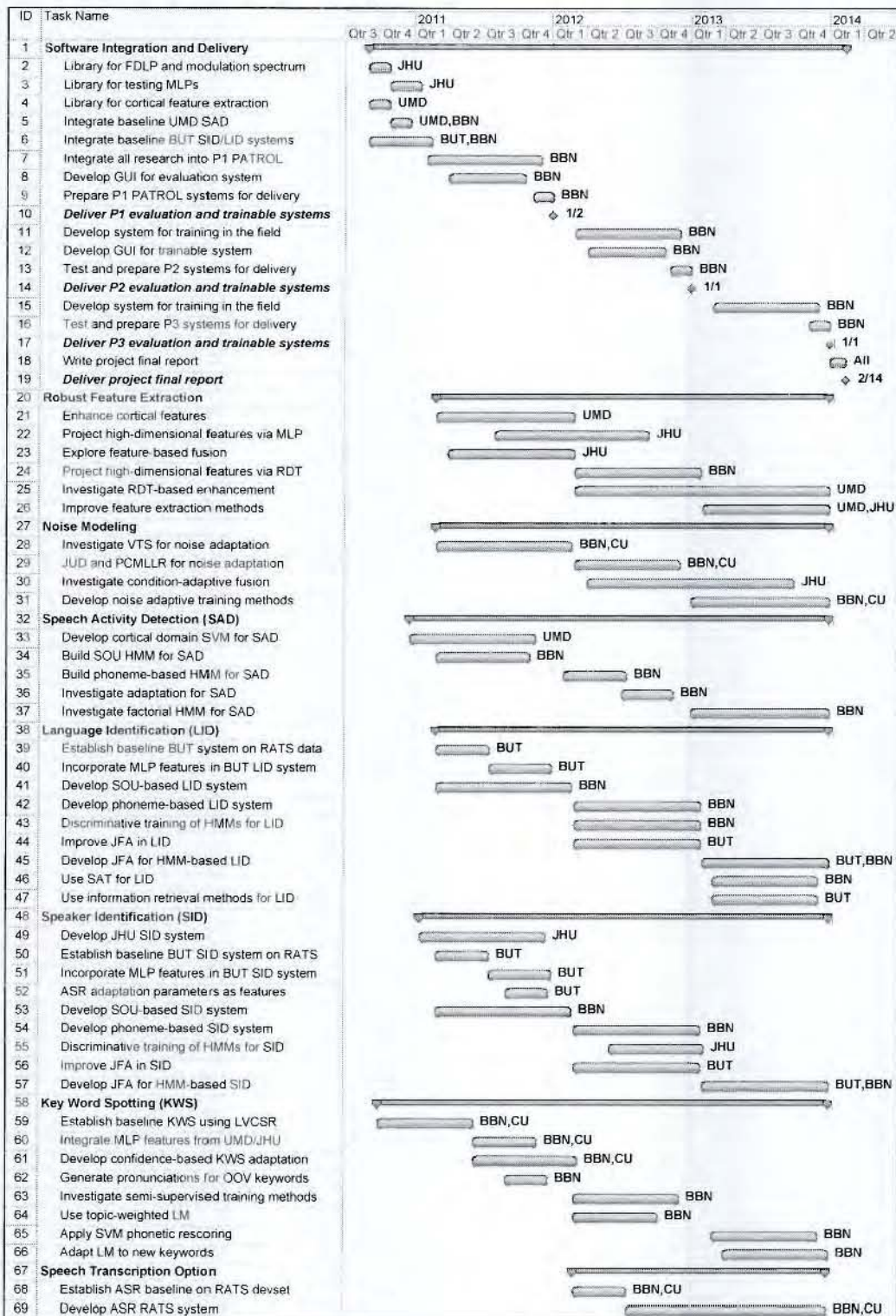
### Intellectual Property Representations

BBN provides its good faith representation that BBN either owns or possesses appropriate licensing rights to all other intellectual property that will be utilized under BBN's proposal for the RATS Program. BBN will obtain from its subcontractors appropriate licensing right to all subcontractor intellectual property that will be utilized under BBN's proposal for the RATS Program.



## 7 Schedule and Milestones

The proposed schedule is given below. Milestones are indicated in *bold italic* font.





## 8 Personnel, Qualifications, and Commitments

For the very challenging problem of performing automatic recognition of language, speaker, and speech in real-world high-noise environments, we have assembled a team ("PATROL") that is equal to the task. Starting with the overall leadership of an experienced staff at BBN, we have added four sites that have unique world-class capabilities in all areas of this project. Dr. John Makhoul at BBN will serve as PI for this effort. He will be assisted at BBN by Rich Schwartz as Technical Lead and Spyros Matsoukas as co-PI and Technical Manager, both of whom played major roles in the DARPA EARS and GALE programs, which required leading multi-site teams working across geographically diverse sites. Makhoul, Schwartz, and Matsoukas will work with staff at BBN with wide and deep experience in all areas of recognition problems in speech, as well as the building and deployment of multi-media operational systems.

Because of the importance of dealing with high noise in this project, we have added three sites that have made significant contributions to the problem of speech in noise. Dr. Hynek Hermansky of Johns Hopkins University brings years of experience in developing robust methods of speech processing, many of which have been adopted by researchers worldwide. Dr. Shihab Shamma of the University of Maryland has developed a unique multi-dimensional cortical representation of sound that he has used successfully to combat many types of unwanted signals and noise in speech. Professors Mark Gales and Phil Woodland of Cambridge University, who have developed one of the top-performing speech recognition systems in the world, have also made novel contributions to the modeling of noise in speech, to the use of discriminative training and adaptation (which will be important to dealing with noise), and in performing speech recognition and keyword spotting under noisy conditions. To round out the team, we have added Drs. Lukas Burget and Pavel Matejka of the Brno University of Technology (BUT), who have developed world-class language ID and speaker ID systems that have competed successfully in NIST evaluations, year after year, scoring top marks in many of the conditions that were competed.

Below is a listing of key personnel in each organization and the unique capabilities they bring to this effort. A table of the time commitments of these key personnel is given in Section 8.6.

### 8.1 BBN

Key personnel at BBN include John Makhoul, Richard Schwartz, and Spyros Matsoukas. (As the need arises, we will draw upon the extensive reservoir of expert advice from other senior staff at BBN, including Herb Gish, Owen Kimball, Man-hung Siu, Walter Andrews, Long Nguyen, Rohit Prasad, Amit Srivastava, and Prem Natarajan. These staff's expertise spans the full range of the RATS applications, as well as the integration and delivery of operational speech systems.)

**Dr. John Makhoul**, a Chief Scientist at BBN, is also an Adjunct Professor at Northeastern University, where he supervises students who do their graduate thesis work at BBN. He has led projects in various aspects of speech and language processing, including speech coding, speech synthesis, speech recognition, speech enhancement, speaker recognition, artificial neural networks, optical character recognition, language understanding, information retrieval and extraction, human-machine interaction using voice, and machine translation. He has served as PI on numerous DoD speech and language projects, including the DARPA EARS speech recognition project and the ongoing DARPA GALE machine translation project, which includes work in speech recognition from Arabic and Chinese. Makhoul's involvement in speech



enhancement goes back to a DoD project in 1979 to enhance the quality of noisy speech. In 1989, he led a National Council panel that produced a study about the Removal of Noise from Noise-Degraded Speech Signals. Makhoul is a Life Fellow of IEEE and a Fellow of the Acoustical Society of America. He has received a number of IEEE awards for his technical accomplishments, including the 2009 IEEE James L. Flanagan Speech & Audio Processing Award and Medal "for pioneering contributions to speech modeling."

**Richard Schwartz**, a Principal Scientist at BBN, is a recognized expert in various aspects of pattern recognition, including speech recognition, HMM modeling, probabilistic language modeling, speaker verification and identification, speech recognition in noise, fast search, neural networks, robust multilingual OCR, and, more recently, machine translation. He has been technical lead or senior technical advisor on numerous DoD projects in those areas, including the EARS and GALE projects mentioned above. Schwartz was the chief designer of the BBN Byblos speech recognition system, which continues to form the backbone of all our state-of-the-art speech recognition and keyword spotting work. The Byblos system has been one of the consistent top performers in official government speech recognition evaluations. Among his many original contributions, Schwartz was the first to place speaker identification on a firm statistical basis in 1982; in 1990 he invented the N-best algorithm, an efficient method for computing the N-best sentence hypothesis in speech recognition; and in 1993 he was the first to demonstrate large-vocabulary, continuous speech recognition in real-time on COTS hardware, using a new search algorithm he had invented. He was a primary contributor to the BBN work on speech enhancement in 1979 – work that continues to be referenced to this day. He designed and supervised the implementation of the first speaker verification system that used hidden Markov models; the system was deployed at BBN in 1986. He has made numerous contributions to speaker change detection, speaker clustering, and search algorithms for 50x faster than real-time keyword spotting.

**Spyros Matsoukas**, a Lead Scientist at BBN, has been working on various aspects of statistical modeling in speech recognition since 1996. He was the top technical contributor in the DARPA EARS project and has taken over the leadership in the continued improvement of the BBN Byblos speech recognition system. Among his original technical contributions is the development of speaker-adaptive training techniques for speech recognition – which is now used in a number of research organizations worldwide – and segmental trajectory modeling for speech recognition. He has also done significant work in the detection of speech in the presence of noise and music, the detection of changes in the acoustic channel and the speaker, and speaker clustering. Matsoukas was the direct supervisor of the PhD thesis of one of our Northeastern University research assistants whose paper with Matsoukas on region-dependent feature transforms for speech recognition won a Spoken Language Processing Grant best paper award at the 2006 IEEE ICASSP conference. More recently, Matsoukas has been in charge of the machine translation work under the DARPA GALE project, which has borrowed a number of statistical modeling techniques from the speech area. A paper resulting from work that he led on system combination in machine translation won the best paper award at the 2007 Association for Computational Linguistics conference on Human Language Technology. In addition to overseeing the technical work of the machine translation work at BBN, he also manages the technical interaction with our subcontractors and is responsible for integrating the work of all sites to produce the highest possible quality machine translation from Arabic and Chinese (speech or text) to English. Matsoukas' excellent cross-site technical leadership skills will serve us well in this proposed work.