Spoken Language Digital Libraries: The Million Hour Speech Project

James K. Baker

Director, Center for Innovations in Speech and Language, Carnegie Mellon West, Moffet Field, CA, USA 94035

***Abstract* — The Center for Innovations in Speech and Language (CISL) at Carnegie Mellon University has launched a grand challenge project to collect and annotate at least one million hours of recorded speech. To collect and analyze such a large corpus, CISL is seeking the help and cooperation of affiliated research centers in many countries. The Million-Hour Corpus is designed to support research into innovative methodologies in knowledge representation and knowledge acquisition applied to speech recognition and synthesis, for which current data corpora have been inadequate. The new methodologies and the strong international cooperation will in turn enable the analysis and utilization of large corpora in ways that heretofore have not been possible. If this joint effort is successful, not only will it lead to improvements in speech systems, but it will make such systems available in many more languages and may even provide the catalyst for a paradigm shift in acoustic modeling.**

***Index Terms* — Socratic Agents, Knowledge Acquisition, Multiple Classifiers, Modular Architecture, Speech Manifold, Hidden Phonetic Sequences, Multi-Stage Recognition, Distributed Learning.**

I. Introduction

The Center for Innovations in Speech and Language (CISL) at Carnegie Mellon University in Mountain View, CA has launched a grand challenge project to collect and annotate at least one million hours of recorded speech. The goal is to collect a substantial amount of data in each of at least 100 languages. To reach this goal, CISL is seeking the cooperation of affiliated universities and research centers in as many different countries as possible.

Why collect one million hours of speech? The purpose of the collection is to support research in improved methodologies for speech recognition and synthesis and the development of advanced speech and language technology in many languages. The Million-Hour Corpus will provide an unprecedented source of data to build models and to study acoustic phonetics to a degree never possible before. The Million-Hour Corpus will provide a unique resource not only because of its sheer size, but also because of its depth, its breadth and its density. In addition, subsets of the collection will be specifically designed to study specific questions that have been difficult to answer with existing corpora.

II. Implications of Size

Several different approaches will be used to collect large, diverse corpora of speech. Each of these approaches by itself may be able to collect as much as a million hours of speech or more, so the overall corpus may eventually comprise a total of several million hours of speech.

Some of the approaches for collection will be designed to collect data that satisfies particular characteristics. For example, a particular goal is to collect data from a wide range of languages, specifically to have at least 100 languages be well represented. Ideally, there will be at least 10,000 hours of speech in each language. Another design characteristic is to have enough data per speaker to be able to build a complete, detailed model of the individual speaker's voice either as a voice model for synthesis or as a speaker-specific acoustic model for speech recognition. This is estimated to require 10-100 hours per speaker.

Other collection approaches will be aimed at collecting data representing a diversity of speaking styles. Thus, the corpus will include both read speech and spontaneous speech. It will include both careful speech and casual, conversational speech. The corpus will include telephone speech as well as higher-bandwidth microphone speech.

To collect such a large corpus, which is nearly two orders of magnitude larger than is presently available, will require several things. It will require international cooperation and broad participation. It also will require that the cost per hour of collected speech be kept as low as possible. In particular, the cost of human transcription and human translation will in general be avoided. Instead, new training and learning algorithms will be developed that are more robust than conventional unsupervised training, but that do not require human-supplied or human-verified labels or transcriptions. In addition, although it is only one speaking style, a large amount of read speech will be collected, because it avoids the cost of transcription.

The Million-Hour Corpus will enable radically new methodologies to be developed, and its effective utilization will require such methodologies. A prime motivation for making such a large increase in the

amount of speech data available is to support such radical changes. It would be extremely disappointing if the Million-Hour Corpus were to be used merely to do more of the kinds of things that are already being done. Its value will be much greater if it is the catalyst for a fundamental change in paradigm that uses this large corpus of data to acquire new kinds of knowledge.

To better understand the proposed new methodologies, the current methodologies must first be placed in a broader context. For the moment, focus on the current methodologies in speech recognition. The predominant method used in current leading speech recognition systems is to model speech as a hidden Markov process [1]. The predominant method for training the acoustic models is to use the Baum-Welch algorithm [2] – [3], which may be viewed as an instance the EM algorithm [4], which is a general statistical technique for training models of a stochastic process with hidden random variables.

In the broader context of paradigms for knowledge representation and knowledge acquisition, the EM algorithm is a method for *training* the parameters of a specified model rather than a method for *learning* new kinds of knowledge. Its use in training acoustic models for speech recognition is unusual because there are tens of thousands or even millions of model parameters whose values are estimated. However, the process is still essentially a process of parameter training. Recent innovations, such as corrective training or discriminative training, are also instances of parameter training, merely with different optimization criteria. More general stochastic process models, such as dynamic Bayesian networks, are also trained using the EM algorithm, and are still instances of parameter training.

For certain specialized tasks, leading speech recognition systems do use other, fundamentally different methodologies, for example, clustering of distinct triphone contexts [5] – [6] or partitioning the acoustic space in estimating a Maximum Likelihood Linear Regression (MLLR) [7] – [8] for adapting models to a given speaker. In principle, clustering or partitioning could be used to learn new structural knowledge. In practice, the clusters or partitions that are learned in most current applications in speech recognition are arbitrary groupings that are used primarily to permit statistical smoothing for better parameter estimation. The clusters or partitions are not designed to try to represent or learn new knowledge. Experiments have shown that, although this smoothing leads to significant improvement in performance, the overall performance of the system depends very little on the particular set of clusters that are used or the knowledge that they might represent.

Moreover, the current clustering and partitioning implementations are not tied to a fundamental knowledge representation and knowledge acquisition process. In particular, the triphone clustering is not associated with a process of learning basic knowledge of allophonic variation as a function of context. MLLR speaker adaptation is not associated with a process of learning systematic pronunciation variations as a function of dialect.

Some studies have used techniques based on more sophisticated knowledge of speech production or speech perception. For example, signal processing and pattern recognition processes have been derived from psycho-acoustic studies and neurophysiology. Speech production has been represented by methods combining knowledge from speech synthesis and speech recognition [9]. Hidden articulatory modeling performs analysis utilizing the continuity constraints of speech production. However, development and utilization of these techniques have been hampered because the standard speech recognition training corpora have been designed primarily with standard hidden Markov process modeling in mind. In particular, there is not enough data per speaker to build a detailed voice model. Moreover, these and other advanced models would also benefit from better methods of knowledge representation and knowledge acquisition and from algorithms for learning new knowledge.

III. International Cooperation

The size and scope of the Million-Hour Corpus require that this project be a coordinated effort of affiliated research sites in many countries. The data collection goal alone (collecting as much as ten thousand hours in each of at least 100 languages) mandates that there be collection sites in many countries, with access to native speakers of many different languages.

The research project, for which the Million-Hour data collection is merely an enabler, will require even more resources and expense than the data collection effort. It will be highly desirable to have many sites, with native speakers of many languages, also participate in the research project. The Modular Knowledge Architecture, discussed below, is designed to facilitate cooperative, semi-autonomous contributions from many research sites. The proposed knowledge representations and methodologies for the research project allow research and even specific knowledge acquisition to be applied across languages.

It is expected that many, perhaps hundreds, of sites will contribute data to the Million-Hour Corpus. The terms and conditions for affiliate relationships will be flexible and will be designed to meet the needs of both universities and for-profit commercial enterprises in different situations in many countries. In exchange for donated data, contributing entities may receive any of several forms of payback, including: license rights to a larger quantity of data, free or reduced tuition for

training at the Center for Innovation in Speech and Language (CISL) at Carnegie Mellon West, cash payments, or other rewards. It is hoped that many universities, and even many commercial companies, will find it attractive to be affiliates of this program.

There will be a small number of Million-Hour Research Centers. These Centers will be world leading research centers in speech and language technology and will be mirror sites for the Million-Hour Corpus. In addition to the paybacks and rewards received by the smaller affiliated sites, these Million-Hour Research Centers will receive a license to the entire Million-Hour Corpus and will be direct participants in the associated research program. They will be required to contribute a large quantity of speech data covering many languages and to maintain world-class research facilities to support the analysis of the Million-Hour Corpus. They will also provide computer and support services to make the Million-Hour Corpus available and accessible for visiting researchers.

IV. Modular Knowledge Architecture

To support the effective utilization of the Million-Hour Corpus, and the new knowledge acquisition methodologies that it enables, a new architecture, called Modular Knowledge Architecture, is proposed. The Modular Knowledge Architecture supports existing modularity in knowledge representation, such as the separation of acoustic modeling and language modeling. It also supports multi-stage recognition architectures with increasingly complex knowledge represented in each successive stage. However, the Modular Knowledge Architecture goes further. It allows each module in a current system to be replaced by an arbitrarily large collection of cooperating, heterogeneous, diverse but constructively redundant modules.

The Modular Knowledge Architecture supports the Million-Hour speech project in several ways. The Million-Hour speech project will require the cooperation of many research sites in dispersed geographic locations. The Modular Knowledge Architecture will support independent development of many cooperating knowledge-source modules in spite of overlapping knowledge and redundancy. It will allow the smooth integration of many modules developed at separate sites without any module needing to know the details or the internal operations of any other module. The system integration tools will permit the insertion or removal of individual modules without disrupting the operation of the system.

The Million-Hour speech project includes not only the collection of at least a million hours of speech, but also the analysis and annotation of the corpus. To support the processing of this huge quantity of data, the Modular Knowledge Architecture scales to distributed-computing on very large networks. Real-time recognition processing is expected to scale to networks of up to 1,000 computers working together on shared real-time recognition tasks. Knowledge acquisition, especially cooperative learning of new knowledge, will potentially scale to networks of millions computers sharing knowledge cross-speaker and cross-language.

Further aspects of the Modular Knowledge Architecture will become clear in the context of the discussions of some of the proposed new methodologies.

V. Socratic Agents

Socrates asserted that he was wiser than other men only in that he knew that he had no true knowledge whereas other men thought that they were very smart, but actually knew nothing [10]. Just as it is useful to distinguish between parameter training and learning of new knowledge, it will be useful to distinguish between knowledge about speech and knowledge about knowledge itself. Knowledge about knowledge and its limitations will be referred to as *wisdom in the sense of Socrates.*

For modularity, this wisdom will be encapsulated not as self-knowledge such as Socrates had but rather in separate, specialized objects called Socratic Agents. Two different kinds of Socratic Agents will be discussed in this paper. The simplest Socratic Agent is an agent associated with a single decision. The methodology can be applied to many kinds of decisions. For this discussion, the example will be to test a particular training token and to decide whether the label given to the training token is a good one.

The problem is that, for various reasons, there may be errors in the labels associated with the training data. With the Million-Hour Corpus this problem is prominent because it would be prohibitively expensive to have human transcription of all of the data, so less reliable means must be used. Even for read speech, there will be occasional reading errors, and it is too expensive to have human verification that every script has been read correctly. A naïve method of making the training robust against labeling errors is to reject every training token that fails to match its designated label better than some specified threshold. However, correctly labeled training data also deviates due to many sources of variability. If all the tokens that match the current model poorly are rejected, then the models will never learn these sources of variability. The correctly labeled training tokens that nonetheless deviate substantially from the current model are very “good” in the sense that they are important for making the models more robust. Thus, this naïve attempt to make the training robust against labeling errors instead makes the models extremely non-robust or fragile.

Socratic Agents are designed to have wisdom in the sense of Socrates. This wisdom is "knowledge about knowledge" that should be distinguished from the base knowledge being studied. In general, Socratic Agents should be designed to utilize information that is not available in the base process or module. This principle is in contrast to the typical conventional "confidence measure," which measures the confidence that a pattern recognition module has that its decision is correct [11]. Typically, such a confidence measure has the same information that is available to the module in making its original decision. It is literally a measure of self-confidence, not true wisdom. Too often pattern recognition modules are very confident in their decisions, even when they are wrong, like the unjustified confidence of the Greek philosophers whom Socrates interviewed and found wanting.

Thus the Socratic Agent acquiring wisdom about the labeling of a given training token, like all Socratic Agents, should be designed to use information that is not available in the base decision. For the reasons discussed above, it is difficult to decide whether the training label is "correct." Instead, the test is modified to a criterion that at first appears even more difficult to check, but that is even more directly relevant to the task. The modified criterion is whether using the training token with the specified label *improves the future performance* of the system.

Using the future performance of the system as a criterion at first appears to require not merely the wisdom of Socrates, but the omniscience of the Delphic oracle. However, we can use a trick. The decision as to whether a given training token is "good" is delayed. To measure the future performance resulting from using the given training token, two copies are made for every model that is affected by the operation of training on the given token. For one copy, the training token is not used. For the other copy, the training token is used. Based on other criteria that will not be discussed here, one of the two copies is selected to actively participate in the recognition process. However, in separate post-recognition analysis, the other copy is substituted and comparative recognition performance is measured.

The problem of deciding whether the given training token is "good" is converted into a standard problem in sequential decision theory [12]. The null hypothesis is taken to be that the two copies of the models (with and without using the given training token) are equivalent in net performance. Comparative performance statistics are accumulated until the null hypothesis can be rejected at a specified level of statistical significance (or until it is determined that further testing is not cost effective).

If the null hypothesis is eventually rejected in favor of the models that used the given training token, that information is fed back to the annotation database and the given training label is marked as "good" (in the sense that training using the label produces improved performance, which is not quite synonymous with the label being "correct"). If multiple models are affected by the given training token, optionally multiple sequential decision tests may be set up. For example, in pronunciation modeling or dialect learning, the word label may be correct, but the pronunciation used by the given speaker may be different from any pronunciation currently in the dictionary. Then separate tests may be set up for the word label (which is not yet known to be correct) and the phoneme labels from the dictionary (which may be incorrect even if the word label is correct).

V. Socratic Controllers

Another form of Socratic Agent may be used to support the Modular Knowledge Architecture. Within the Modular Knowledge Architecture, there will be a Socratic Agent controller for every collection of modules cooperating on a shared recognition task. This Socratic Controller will be responsible for controlling the collective training of its component modules to jointly optimize their shared objective. The controller will invoke the training or learning procedures of the component modules, but it will not need to know the internal knowledge representation or the training algorithm of each component module. Similarly, each component module will have no direct knowledge of the presence of particular other modules, much less their internal structures and procedures. However, in the joint training the Socratic Controller will selectively present training data to the component modules, essentially assigning responsibility to particular components for learning particular pieces of knowledge. Thus, as trained in the context of a specific collection of other modules, the models and trained knowledge of a given module will be very different than the same module trained on the same set of training data in the context of a different collection of cooperating modules.

In the general statistics and machine learning literature, there are many algorithms for training systems with multiple classifiers [13] – [14]. However, these algorithms generally assume that the classifiers are all copies of the same basic, simple classifier design. The techniques generally do not apply to collections of heterogeneous complex recognition systems, such as large vocabulary speech recognition systems. On the other hand, there is no agreed standard for knowledge representation among the leading recognition systems, so in most multiple-system recognition experiments the component systems have been trained independently on the specified training data. For adaptive training on the test data, there are no human supplied labels. A successful technique that has been used with multiple systems is cross-adaptation. In cross-adaptation, a

system is adaptively trained using the output of another system as the training labels (without knowing whether or not each label is correct).

The general research on multiple classifier systems overwhelmingly indicates that overall system performance is improved if the degree of diversity is increased among the collection of component classifiers. However, independently training each of the component recognition systems does nothing to increase their diversity or to optimize their joint performance in any other way. Cross-adaptation, which is applied to the test data, actually tends to decrease the diversity. Apparently it works in spite of decreasing the diversity because the alternative is unsupervised self-adaptation, which is notoriously unreliable. Socratic Agents, however, provide a means to jointly optimize the combined performance of the collection of modules as well as to robustly train on the test data and on unlabeled or partially labeled training data.

Basically, the delayed-decision technique is applied. For each training token, a Socratic Agent is set up for each component module to evaluate the decision of whether to train that module using the given training token. On test data or partially labeled training data, a potential training token can be set up with each of the labels returned as a recognition label by one or more of the component modules. On many training tokens, all of the component modules will successfully reject the null hypothesis in favor of the using the training token. To increase diversity, however, not all component modules will be selected for training using that token. Instead, performance statistics will be accumulated for each component estimating the amount of performance improvement (relative to the incremental amount of resources used, if any). Only a relatively small number of the best components will be selected to use a given training token. Note that the performance improvement is always measured as the improvement in the overall system result, not in the label made by the component module by itself. Thus the Socratic Controller jointly controls the training of the collection of component modules; it selectively assigns responsibility for each training token to increase diversity; and it optimizes the combined performance of the overall system not the performance of individual components.

The Socratic Controller also coordinates the component modules and computes the combined result in any recognition task. The wisdom, or knowledge about knowledge, that must be used in this case is to know how reliable each component module is *in this particular case,* that is, on the given task with the particular set of given data. To have effective wisdom, the Socratic Controller uses information that is not available to the component modules individually. In this case, the information available to the Socratic Controller includes the output results of the component modules.

Note: a system of modules that use each other's output as input data are interdependent and, hence, are not modular under the definition of modularity for the Modular Knowledge Architecture. Such a system of modules would be treated as one large, complex module within this architecture. Of course multiple copies or variants of this single, complex module could be created and could be trained for diversity as described above. Thus, the complex module can easily be imbedded in an even more powerful, multiple module system within the Modular Knowledge Architecture, so the techniques of this section can still be applied.

The output results of the component modules include each component's estimation of the best matching label. Optionally, they include a ranked list of the best matching labels and scores for each label. These outputs and the union of all the input data to the component modules form the vector of input data to the Socratic Controller. In particular, the Socratic Controller can count how many component modules agree; it can test whether or not two particular component modules agree; it can compute functions of the raw input data; it can form unions and intersections of the lists of best matching labels; and it can do many other computations to test the reliability of the component predictions.

In the simplest methodology, the output of the Socratic Controller may be considered to be a bit vector with one bit for each component module. The bit for a particular component could be one if the Controller believes that the best-result output of the particular component should be used in this particular case and could be zero if the Controller believes that the output of the particular component should be ignored. In this simple scheme, the combining function could be a simple voting scheme with fixed weights.

In a slightly more complicated methodology, the vector could have three values for each component: "believe," "don't believe," and "don't know." In more complex methodologies, the combining function could be an arbitrarily complex, parametric function of the available data. In such a case, a requirement is that it must be feasible to determine the optimum values of the parameters in the parametric combining function in the case of training data, that is, when the correct answer is known.

In all of these cases, the task that the Socratic Controller must do is itself a pattern recognition problem that can be solved by conventional pattern recognition methodology. The difference is that the pattern recognition process for the Socratic Controller is not trying to directly recognize the correct output label. Rather it is operating at a meta-level (the wisdom level rather than the knowledge level) and is trying to represent knowledge about the knowledge of the component modules. However, there is no need to invent new training and recognition techniques just for

the Socratic Controller's recognition problem. Any of the wide selection of standard pattern recognition techniques may be used. The Controller's pattern recognition problem as different inputs and different outputs from the component modules, and it can be trained independently.

VII. Learning New Pronunciations

Socratic Agents enable the *learning* of new knowledge as well as the training of models. An example for such learning will be described with respect to learning new pronunciations. There are many reasons that a given speaker's pronunciation of a particular word may be different from the pronunciation that is specified in the dictionary. One reason, variation among the allophones of a single phoneme, will be discussed in another section. Other reasons include errors in the dictionary, words that have multiple pronunciations, dialectal variation, foreign accents, and individual speaker idiosyncrasies. For any of these sources of change in the pronunciation, it might be appropriate to add a new pronunciation to the dictionary, or to make some other change in the knowledge representation of possible and probable pronunciations. However, there are also other sources of variability in the acoustic features observed with an instance of a word besides a change in the underlying pronunciation, so it might not be appropriate to make a change in the dictionary.

When an apparently new pronunciation is observed, the question then is whether the potential new pronunciation should be represented. It seems that this question is regarded to be difficult to answer, or at least tricky in the sense that answering it wrong not only fails to improve performance, but instead can make it worse. Most systems, therefore, don't even attempt to automatically add pronunciations to the dictionary much less to construct or modify more complex representations of pronunciation knowledge. Some experiments have even shown degradation in performance when new pronunciations that have been verified by human reviewers are added to the dictionary.

A Socratic Agent, however, is explicitly designed for making such a decision that may affect the future performance of the recognition system. This is simply another application of the delayed-decision Socratic Agent that was proposed above for deciding whether to use a given training token. A slight modification is made in the decision criterion because representing a new pronunciation will not only affect performance, but also may require additional computational resources. Thus the situation is asymmetric. The base hypothesis is that the new pronunciation fails to improve performance by more than its marginal cost. The new pronunciation should only be accepted if this hypothesis can be rejected at a statistically significant level.

The selection of future performance evaluation samples will depend on the suspected or hypothesized source of the variability in the pronunciation. If the new pronunciation is to be tested as a new entry in the dictionary, then the new pronunciation is be tested for all hypothesized word strings that include the given word, for all future recognized utterances, for all speakers of the language, until enough evidence is accumulated to accept or reject the null hypothesis. If the new pronunciation is to be tested as a variation due to dialect or foreign accent, then the new pronunciation may be accepted even if it only improves performance for an identifiable subset of the speakers. On the other hand, for variation due to dialect or accent, the new pronunciation might be represented as an instance of a new pronunciation-modification rule that will also apply to other words in the dictionary. In that case, the Socratic Agent will accumulate performance statistics across all the affected words.

VIII. Learning Invariants

The Million-Hour Corpus will be designed to have a large number of speakers for which there is a substantial amount of recorded speech for each speaker (at least 10 hours per speaker). This design decision is intended to promote in-depth scientific studies of the characteristics of each voice. It is not based on a focus on speaker-dependent rather than speaker-independent speech recognition tasks. Indeed, the corpus will also be designed to have a large number of speakers (at least 100,000), so it will also be an unprecedented resource for speaker-independent modeling. With a substantial quantity of speech from each of a large number of speakers, several new methodologies for knowledge acquisition will also be enabled.

When a human listener who is a native speaker of a language hears a particular sound or phoneme in that language, the listener immediately perceives an instance of that phoneme, regardless of whether the speaker is a man, a woman or a child. Unless highly trained in acoustic phonetics, the listener generally isn't even aware of the great differences in the physical sound due to the gender, age and other individual characteristics of the particular speaker. That is, the human speech perception system automatically and subconsciously computes invariants that make multiple instance of the same phoneme sound alike and instances of distinct phonemes sound different.

The task of speech recognition would be much easier if computers could be instructed how to compute these invariants. However, attempts to compute such invariants have not yet met with much success. One of the problems in trying to learn the right functions to compute these invariants is that there is not enough data of the right kind to be able to learn very complex

functions. The Million-Hour Corpus and its planned annotations are designed to support research in areas such as the search for invariants. For example, there will be tens of thousands of hours of scripts for which the same script is read by many speakers. For a large part of the Corpus, there will be enough data per speaker to allow a detailed model to be built for each individual voice. Such detailed models will enable the inter-speaker variability to be separated from the intra-speaker variability, which will facilitate the construction of invariants that apply across speakers. In addition to word and phoneme alignments, the Million-Hour Corpus will also be annotated with language-independent phonetic labels. Among other things, this annotation will help in finding invariants that apply across languages.

Ironically, the search for invariants is sometimes justified by the claim that finding invariants will help reduce the amount of data that is needed to train acoustic models in a new language. Indeed, invariants may be valuable for this as well as for other reasons. However, it is clear that once the Million-Hour Corpus is available, the problems of producing acoustic models for any of the 100 languages will already be greatly reduced.

IX. Language-Independent Phonetic Labels

The Million-Hour Corpus will be an annotated corpus, with the annotations largely produced by automatic processes. It is expected that the annotations will be continually updated and improved as additional analysis is done. For all the data, automatically generated labels will be available from running recognition on the data. For read speech, the prompting scripts will be available. Human-corrected transcripts will be available from speech recorded in interactive applications that support on-line error correction. All of these scripts and transcripts will be marked for reliability using training token, delayed-decision Socratic Agents. These transcripts will be used to compute word alignments and phoneme alignments based both on the base dictionary and on expanded dictionaries incorporating new pronunciations developed as described above.

In addition to these language-specific annotations, the Corpus will be annotated with language-independent phonetic labels using the International Phonetic Alphabet (IPA). At least two methodologies will be used to produce these phonetic annotations. One method will be simply to build a pronunciation dictionary in each language based on the IPA rather than on a language-specific phoneme set. Since the mapping from phonemes to IPA symbols is not unique, there may be many possible pronunciations in the IPA notation that correspond to the same phoneme sequence. Automatic procedures based on Socratic Agents will test and verify each candidate IPA pronunciation.

Another methodology will be to represent the phonetic sequence as a hidden stochastic process (in addition to, and dependent on, the phoneme sequence). This technique allows the explicit representation of allophonic variation and of the context-dependent probabilities of the allophones. The phonetic sequence cannot be represented as a hidden Markov process, because the probability of each allophone depends on the underlying phoneme sequence. However, it is a well-defined, well-structured hidden stochastic process, so the models can still be trained by the well-known EM algorithm, with a computation that is very similar to the one used to train models of a hidden Markov process.

Representing the phonetic sequence as a hidden stochastic process also has other implications and potential benefits. In particular, it represents some of the variability explicitly as probability distributions over the finite set of phonetic labels. This variability would otherwise need to be represented by continuous probability distributions over the acoustic feature vector space (such as by mixtures of Gaussian distributions). Representing the variability by explicit discrete probability distributions over finite alphabets will facilitate representing and learning the context dependencies of these probability distributions. Representing the probability distributions of the acoustic features as dependent on language-independent phonetic elements rather than language-dependent phonemes will facilitate applying acoustic model knowledge learned in one language to other languages.

X. Speech Manifolds

Because of the physical and physiological constraints on the positions and motions of the articulators, actual speech is limited to a small portion of the large, high-dimensional space of acoustic feature vectors. However, the constraints and their effect on the acoustic features are complex and highly non-linear. Therefore, the proper mathematical structure for representing the embedding of the set of acoustic feature vectors corresponding to speech in the overall high-dimensional space of acoustic feature vectors is a manifold, that is a lower-dimensional, twisting curved surface embedded in the higher dimensional space. Actually, because speech is modeled as a stochastic process, it is represented by probability distributions in acoustic feature space. The approximating manifold being discussed here is a lower-dimensional surface that describes most of the variance in the speech sounds at a given location on the manifold.

This speech manifold is mostly smooth. That is, at most places on the speech manifold the manifold can be approximated in a local region by a tangent plane of the same dimension as the manifold. For a large number of speakers, the Million-Hour corpus will have ten to one hundred hours of speech. This is enough data per

speaker to build a high-quality speech synthesis voice model. That is, it is enough data to train models that capture the detailed characteristics of the individual voice. Interpreted another way, it is enough data per speaker to build an accurate description of the speech manifold of an individual speaker.

In addition, each speaker's manifold is labeled by the annotations that have been associated with the speech data that has been mapped to the manifold. For any two speakers, a mapping consistent with the labeling can be constructed between the two speech manifolds. Generally, this mapping will be smooth. Therefore, for two very similar speakers the mapping can be approximated in any small region by a linear mapping. Thus, at any point on the manifold, the mapping can be factored into a component in the tangent plane of the manifold and a component orthogonal to the tangent plane. The orthogonal component will also generally be a lower-dimensional subspace and will be useful in constructing speaker-independent invariant functions.

A new methodology for speaker-independent modeling will be developed based on the single-speaker speech manifolds. For any unknown speaker (or any known speaker with a limited amount of training data), a selection is made of a small number of speakers with similar voices (or voices that are similar after transformation by any other available speaker normalization transform). Among the similar speakers the speech manifold mappings can be represented locally by lower-dimensional linear transformations. Therefore, interpolation between the speech manifolds can be represented with a small number of parameters. Thus, a relatively small sample of speech from the new speaker can be used to estimate a new speech manifold that is an interpolation among the manifolds of a small number of similar speakers. Phonetic labeling of speech manifolds may even enable speaker-independent interpolation across speakers of different languages.

Although the mappings between the speech manifolds of individual speakers are generally smooth, there are certain significant exceptions. These exceptions are not to be ignored, but rather are worthy of study as another source of knowledge. For example, two speakers with different dialects will often result in a speech manifold mapping that is discontinuous for certain words or phonemes. For dialect studies, the intent is to study the discontinuities rather than to avoid them, so phonemic labeling will generally be more useful than phonetic labeling. Detecting and characterizing these discontinuities will enable the development of dialect transformation rules and dialect-specific pronunciation dictionaries.

XI. Verification by Synthesis

A frustrating result that often occurs in speech recognition research is that new methodologies that clearly have knowledge that is lacking in previous systems often fail to improve the system performance. One problem is that early implementation of new methodologies that are radically different from the older methodologies are often fragile. Although they may contain significant new knowledge and may work much better in some circumstances they may fail catastrophically in other circumstances.

Another problem is that methodologies and modules that use more sophisticated knowledge representation are likely to have incomplete coverage in the knowledge represented. Either of these problems may cause an otherwise promising new methodology to not perform as well as older methodologies that have been developed and improved over a long period of time and trained on a large quantity of data. The most innovative of new processing methodologies may suffer from both of these problems.

When a large number of redundant cooperating modules are used, the Modular Knowledge Architecture supports the successful use of such innovative, but potentially fragile new methodologies.

An example of such a methodology is to model speech production. Clearly, if it were possible to accurately compute the positions of the speech articulators from measurements of the speech waveform, the speech recognition problem would largely be solved. However, it is much more difficult to compute the configuration of the vocal tract or the positions of the articulators from the speech waveform than to do it the other way around.

Computing the waveform from the configuration of the vocal tract may be viewed as a form of speech synthesis. The proposed methodology for speech recognition is to do verification by synthesis. The word "verification" refers to the fact that the intention is to perform the synthesis computation as part of the comparison of a small number of hypothesized word strings in the final stage of the recognition process. Verification by synthesis differs from the more difficult "analysis by synthesis" because verification by synthesis is synthesizing a specific given hypothesized word and phoneme sequence, rather than trying to analyze all possible sequences.

Notice that in verification by synthesis there is no need to perform the complex and ill-conditioned computation of the vocal tract configuration given the waveform. Instead the knowledge of speech production and synthesis is all used in computing a waveform from each of a small number of candidate hypotheses. For each hypothesis, the vocal tract configuration is not computed from the waveform, but rather from the given word sequence which in turn determines the phoneme sequence.

Thus, verification by synthesis uses speech production knowledge in a way that avoids the greatest source of fragility in direct modeling of articulator positions. However, the synthesis model and the knowledge of speech production may still be incomplete. For some sounds, the synthesis may be based on only a crude approximation to the actual physical acoustics of the vocal tract.

That is, the synthesis model may have much better knowledge of some sounds but less accurate knowledge of other sounds. The Modular Knowledge Architecture is specifically designed to be able to make good use of such modules. When included with a set of cooperating modules, including the best of the existing methodologies, a module based on verification by synthesis only needs to improve recognition performance in an identifiable subset of situations.

XII. One-Shot Learning

There is great value in the ability to be able to learn something new from a single example. There is a conventional wisdom that hidden Markov models always require a large number of training examples. However, that conventional wisdom is not true for models of speech as a hidden Markov process. The Markov transition network for acoustic modeling has very few transitions with non-zero probabilities per state. Furthermore, accurately estimating the transition probabilities is not essential. In fact, many systems ignore the transition probabilities within the acoustic networks. Thus, if a given set of models require a large amount of data for training, it is for the estimation of the distributions of the acoustic feature vectors, not for the estimation of the Markov process itself.

Perhaps the conventional wisdom about the amount of training data required to train hidden Markov models dates from the early days, before Markov modeling became dominant, and it was being compared with other modeling methods that at the time required less training data. However, the Markov modeling at that time used finite alphabet labels as the acoustic features, and discrete probability distributions needed to be estimated. It does take a large number of examples to estimate a discrete probability distribution, unless there is a lot of prior knowledge of an underlying parametric structure. It also takes a large number of examples to estimate a mixture of Gaussian distributions with a large number of component Gaussians, which is the dominant method for representing the acoustic feature distributions in the leading modern systems. However, these are not the only methods for representing the probability distributions for the acoustic features. The need for a large number of training examples is not inherent in modeling speech as a hidden Markov process.

A much more appropriate model for the acoustic feature distribution for an element for which there is only one or a small number of training examples is a simple, single multivariate Gaussian distribution. That is, although multivariate, it is not a mixture of Gaussian distributions but rather a single Gaussian distribution.

When the new object to be learned is a sequence of more elementary items, such as a word is a sequence of phonemes, first a base network is created. The base network is a sequence of nodes with an arc connecting each node with the following node. In addition the base network has additional arcs, such as an arc going from each node back to itself and an arc going from each node directly to the node that is two nodes later in the network. The additional arcs allow the network to be matched against other instances of the new object in which a node may be skipped or may be repeated. Note that the nodes in this network do not necessarily correspond to a given segmentation, such as the segmentation of a word into phonemes.

To align the observed frames of acoustic feature vectors to the network, either the acoustic frame sequence can be segmented using standard bottom-up phonetic segmentation methods (then the number of nodes is set to be the same as the number of segments), or a dynamic programming computation may be used to find the segmentation that is the best in terms of a minimum least squares criterion. When the dynamic programming routine is used, the number of segments may be externally specified.

When a second or subsequent instance of the object is encountered, the acoustic feature frames may be aligned to the existing model using standard hidden Markov process alignment routines.

Once the frames have been aligned to the nodes of the network, the Gaussian model associated with each node is created. The means for the new Gaussian model for a given node are merely the sample means for the acoustic feature vectors for the frames that are aligned to the given node.

The variances for the new model need special treatment. Some of the nodes may have only a single aligned frame, in which case there would be no way to compute a sample statistic for the variance. Even when there are multiple frames aligned to a given node, it is preferable not to use the sample variance as an estimate for the variance in the associated Gaussian model. The variability observed in the adjacent frames of a single instance of a new object will generally be much less than the variability from one instance of the object to another, which is what the Gaussian model is supposed to represent.

Instead, the variances for the Gaussian model associated with a given node of the new object are taken to be the same as the variances of single Gaussian models of a similar sound or a weighted average of

similar sounds. For the Gaussian associated with this initial one-shot model created from a single instance, it is not essential to accurately estimate the variances. In fact, it is safe to use initial estimates of the variances that are conservatively larger than the actual variances.

As more instances of the new object are encountered, an empirical Bayes estimation procedure is used in which the variances of the similar sounds are treated like a certain number of observations of squared deviations from the mean for the new model being estimated. The relative weight for the pseudo observations taken from similar sounds compared to the observations of deviations from the mean for the actual instances of the new object is determined by the statistical measure of relevance, based on the bias of the estimate from the similar sounds compared to the sample variance from a small number of instances of the object.

The newly created model is in a kind of probationary status. The single instance from which the model was created might turn out to be atypical of the object being modeled. A Socratic Agent is associated with the new model created by the one-shot learning. This Socratic Agent monitors the future performance of the newly created model and eventually decides whether the new model should be given normal, non-probationary status or whether it should be deleted. The Socratic Agent compares the incremental improvement in overall system performance to the incremental cost of the resources required by the new model.

XIII. Creating New Modules

The Modular Knowledge Architecture supports a large number of modules working cooperatively on each task within the recognition system. In addition to the modules created at system design time, new modules can be created at any time during the operation of the system. A delayed-decision Socratic Agent generally monitors the performance of two versions of a module. For making a yes/no decision, the Socratic Agent accumulates data on the net performance to accept or reject a null hypothesis. However, it often may be the case that one version works better in some situations and the other version works better in other situations.

In such a case, another option besides choosing between the two versions is to use both of them. To implement this option, a Socratic Controller is created to control and train the collection of two module versions. If the two module versions are both candidates to be component modules in an existing collection of modules, the collection is merely expanded to include both of them and they are both put under the control of the Socratic Controller for the existing collection.

In addition to the existing or new Socratic Controller, a Socratic Agent is created to monitor the comparative performance of the system with the existing module set versus the performance of the system with the expanded set of modules. As with new pronunciations and new models created by one-shot learning, the Socratic Agent accepts the expanded module set and rejects the null hypothesis only if the performance improvement is greater than a criterion that depends on the incremental resources required by the expanded model set.

As will be seen in a later section, new modules are also created by incorporating models or other knowledge structures that are shared by other systems in a multi-system network.

XIII. Forgetting

When new pronunciations, new models or new modules use up resources, the delayed-decision Socratic Agents accept or reject the new objects based on a criterion that takes account of the marginal cost of the probationary new object in terms of incremental computational resources. The imputed marginal cost of a given amount of a scarce resource can be adjusted to control the rate of creation of new objects. In particular, the marginal cost can be set very high as resources become increasingly scarce so that the creation of new objects does not exhaust the available resources.

Note that it is assumed that the training and learning (and the hypothesis testing done by Socratic Agents) is performed off-line, non-real-time, or on a larger platform than the recognition task, so more resources are available for the training and learning operation than during active recognition. However, even assuming there are additional resources, the resources available for the training and learning operation also could become scarce. If so, then a cost/benefit criterion is also imposed on the number of Socratic Agents active at any one time. Note that, because the training and learning is off-line and non-real-time, the creation of a potential Socratic Agent does not need to be foregone entirely, but may be postponed until other Socratic Agents have finished and freed up resources.

However, even with Socratic Agents controlling the rate of creation of new objects, the procedures described so far will result in a gradual, but never ending, increase in the number of objects (models and modules). However, as new models and modules are created, older models and modules might no longer contribute as much to overall system performance as when they were previously evaluated.

The Modular Knowledge Architecture supports the removal of existing modules as well as the creation of new modules. In addition, there may be multiple units providing alternate representations of the same object within a knowledge representation structure, for example, multiple pronunciations for the same word in the pronunciation dictionary. Any such alternate representation of the same object, or any module within

a collection of cooperating modules is potentially redundant.

As a continuing operation during the learning operation, a random subset of the potentially redundant elements is selected for cost/benefit performance testing. For each element selected to be tested, a Socratic Agent is created to measure the net contribution to the overall performance compared to the incremental resources required. The Socratic Agent performs a sequential decision test of the same kind used to accept or reject new models and modules. Thus, the number of models and modules is continually expanding and contracting, always improving the overall performance-to-resources ratio within the limitation of the resources available.

XIV. Multi-system Architecture

The Modular Knowledge Architecture scales to large networks of computers cooperating on shared recognition tasks. For recognition, a collection of modules may include modules running on different computers within the network. Because of the modular design, the component modules do not need to be explicitly aware of whether or not other cooperating modules are running on the same computer. However, the Socratic Controller in a large network configuration should be aware of the placement of the component modules within the network, so that the Socratic Controller can minimize the communication load on the network.

In general when there are a large number of component modules, even within a single computer, the Socratic Controller will choose a sparse subset of the component modules to be active in a given situation. In a network configuration, the Socratic Controller is a distributed object with a Sub-Controller acting semi-autonomously running on each computer. In a given recognition situation, the data is broadcast to all the computers containing component modules. Each Sub-Controller only activates a sparse, possibly empty, subset of the component modules running on its computer. Results only need to be communicated for the active component modules.

Turning from recognition to learning, a large multi-system configuration supports a major change in the learning paradigm. Note that, because knowledge is shared across languages as well as across speakers, there may be hundreds of thousands or even millions of computers working on shared learning tasks.

In this multi-system learning paradigm, each component system continually creates Socratic Agents to evaluate the performance of new and old modules and models, based on the evaluation data that is locally available. The most promising models and modules are communicated to a limited number of other systems for further evaluation.

When a component system receives a shared model or module from another system, it creates a Socratic Agent to evaluate the performance contribution of the shared model or module. This performance contribution is measured on the evaluation data available to the particular component system in terms of the incremental contribution made in the context of the other models and modules that the particular component system already has.

The incremental contribution to performance of a given module may be large on a system that has no other module with similar knowledge and may be much less on another system that already has other modules with similar knowledge. It is necessary to avoid instability in the evaluation process due to idiosyncratic differences in the current set of models and modules. Accordingly, each system is given a shared base set of models and modules. This base set cannot be modified by an individual component system, nor can any of its elements be deleted or deactivated. The incremental performance evaluation of any newly created or newly imported module will be in the context of a collection of modules that always at least include this base set.

Once a component system has finished evaluating the performance of an imported module or model, the evaluation result is reported back to the system that originated the imported module. The most promising imported modules are passed on for further evaluation in additional component systems. The system originating a given shared module or model accumulates statistics of the evaluation results reported back by other component systems. Eventually, the originating system accumulates enough data to accept or reject the given model or module for inclusion at the next update of the base set of models and modules.

XV. Data Sources and Collection Methods

For the data collection efforts, the first priority will be to collect a large quantity of data of acceptable quality at as low a cost as possible. It is highly desirable that collectively the data represent a wide diversity, especially with regard to speaking style, genre, and topic domain. Also important, but of lower priority, will be diversity with respect to dialect, age, and other individual characteristics of the speakers.

To meet the goals of diversity at low cost, it is anticipated that three major sources will be used for speech data collection: read speech, recordings of radio and television broadcasts, and recordings of spontaneous speech of individuals who have agreed to have their conversations recorded. A special case within the third category will be speech recording in the course of usage of an interactive computer application that includes speaking as a natural part of the application.

Although read speech is a small fraction of existing corpora of speech available to the research community, in many situations it is the least expensive way to obtain speech for which there is a script or transcript. In many countries the wages for literate, educated people is still very low compared to wages in developed countries, especially if no specialized technical training is required. Therefore, the cost per hour of read speech is very moderate and as much as a million hours of speech may be obtained for a reasonable budget. The labor time, and therefore the cost, of human transcription of the recorded speech is an order of magnitude greater, and the transcription task requires trained personnel. Therefore, for the Million-Hour Corpus, the plan is to use no human transcription (except to the extent that it is freely available in interactive computer applications), but only automatically computed transcriptions.

The specifications for the read speech will be designed to minimize the collection costs. In particular, the read speech does not need to be collected in a special recording studio, but may be collected in a normal office or home environment. The speech processing methods in the research project must be tolerant of the environmental noise encountered in such recording environments.

Furthermore, there will be no requirement that the readers read without errors, or that they exactly follow the script. There will be no requirement that anyone listen to and verify the correctness of the recording. The speech processing and training methods will need to be tolerant of the reading errors that normally occur in the course of reading long passages of text. Socratic Agents will enable training on such errorful data.

XVI. Fitting It All Together

The Million-Hour Corpus, the broad international cooperation, and the new methodologies are co-dependent. They enable and facilitate each other, but they also depend on or even require each other. They also require other new techniques that are too numerous to discuss in detail.

The collection and analysis of such a large corpus requires the joint efforts of many people. The broad language coverage plus the degree of language knowledge needed for some of the analyses requires the help of native speakers of many languages. New language-independent representations of acoustic-phonetic knowledge will be needed to properly annotate and analyze this corpus. The Modular Knowledge Architecture will allow research sites spread among many countries to productively cooperate without requiring tight central supervision.

Language-independent phonetic labeling and analysis of acoustic invariants will aid in porting knowledge across languages, which will be vital in enabling the development of advanced speech systems in so many languages are a reasonable expense of time and effort.

Socratic Agents will enable training and learning algorithms that are robust against the many known sources of speech variability and of the many new sources that are surely to be encountered in this corpus.

Socratic Agents will also enable training on data that has not been transcribed or verified by humans, which will be essential for annotating the corpus at reasonable cost.

The quantity of data per speaker and the repetition of the same script by many speakers will facilitate the detailed phonetic analysis, the search for invariants, and the construction of speech manifold models. These techniques will in turn facilitate the cross-language acoustic knowledge acquisition.

Socratic Agents applied to design decisions and performance-tuning parameters will be able to automatically generate many complementary modules, the management of which will need the Modular Knowledge Architecture. Collections of these modules, controlled and trained by Socratic Controllers, will provide a rich implementation of the Modular Knowledge Architecture concept.

The Modular Knowledge Architecture with Socratic Controllers will support the creation and use of specialized modules that seek a higher level of expertise on certain problems, but that may not do as well in all situations. Some of these specialized modules will be based on human-supplied knowledge that has not previously been successfully integrated into stochastic modeling based systems. The broad international cooperation will make it possible to get such human-supplied knowledge across the wide range of languages planned in this project.

In general, the Million-Hour Corpus and the diverse set of new methodologies are interdependent and support each other in a set of round-robin mutually re-enforcing chains.

XVII. Summary

A very ambitious data collection effort comprising at least a million hours of recorded speech is described. This Million-Hour Corpus will be the basis for a major international research program aimed at dramatically improving the performance of speech recognition and speech synthesis systems, as well as making these technologies available is many more languages.

Associated with the Million-Hour Corpus are many new methodologies and algorithms, in particular the Modular Knowledge Architecture and Socratic Agents. These new methodologies are both enabled by the Million-Hour Corpus and are the tools by which the Million-Hour Corpus will be effectively utilized.

There are pending patent applications covering many of the methodologies described in this paper. Affiliated research centers will be able to earn royalty-free licenses to these patents by collecting and donating data to CISL.

REFERENCES

[1] J. K. Baker, "Stochastic Modeling for Automatic Speech Understanding," in *Speech Recognition,* edited by D.R. Reddy, Academic Press, 1975.

[2] L. E. Baum, "An Inequality and Associated Maximization Technique in Statistical Estimation for Probabilistic Functions of a Markov Process," *Inequalities*, Vol. III, 1972, pp. 1-8.

[3] L. E. Baum, T. Petrie, G. Soules, and N. Weiss,. "A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains." *Annals of Mathematical Statistics*, 41 164-171, 1970.

[4] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm." *Journal of the Royal Statistical Society, Series B*, 39(1):1--38, 1977.

[5] L. Gillick, Y. Ito, J. Young: "A probabilistic approach to confidence estimation and evaluation," in *Proc. ICASSP* 1997, Vol 2, pp. 879 ff, Munich, April 1997.

[6] K. Beulen, E. Bransch, and H. Ney, "State tying for context dependent phoneme models," in *European Conf. on Speech Communication and Technology*, Rhodos, Greece, Sep. 1997, pp. 1179 -- 1182.

[7] C. J. Leggetter, and P. C. Woodland, "Speaker adaptation of HMMs using linear regression" (Technical Report CUED/F-INFENG/ TR. 181). Cambridge University, Cambridge, UK, 1994.

[8] M. J. F. Gales, "Maximum likelihood linear transformations for HMM-based speech recognition," Cambridge University, Cambridge, UK, Tech. Rep., 1997.

[9] S. Dusan and L. Deng. "Recovering vocal tract shapes from MFCC parameters," *Proc. ICSLP*, 1998, pp. 3087-90.

[10] Plato, *Apologia Sokratous* (Defense of Socrates).

[11] Thomas Schaaf and Thomas Kemp, "Confidence Measures for Spontaneous Speech Recognition". *Proc. ICASSP-97*, Vol. 2, pp. 875-878, 1997.

[12] A. Wald, *Sequential Analysis*, John Wiley and Sons, New York, 1947.

[13] Y. Freund, "Boosting a weak learning algorithm by majority," *Inform. Comput.*, 121(2), 256--285, 1995.

[14] L. Breiman, "Bagging predictors." *Machine Learning*, 24:123—140, 1996.