09/01/2000 — 12/31/06: RESEARCH AND EDUCATIONAL ACTIVITIES

The overall goal of this ITR was to create a strong synergy between speech recognition (ASR) and natural language processing (NLP). At the time this project began, integration of ASR and NLP was in its infancy, particularly for conversational speech applications. Over the duration of this project, two significant things happened. First, through the parallel efforts of DoD-funded research, community-wide focus on converstaional speech was achieved. Progress was impressive as error rates on tasks such as Switchboard and Call Home English decreased from 50% to 10%. ASR technology was now producing transcripts that were useful to NLP systems, and could support information retrieval applications involving important quantities such as named entities.

Second, NLP research began to focus on the problem of parsing speech recognition output, which lacks punctuation and formatting that was previously considered crucial to high performance parsing. This latter issue was the main focus of this ITR, and to some extent served as a beacon for the community. We produced resources that were extremely valuable, such as the extensions to the Penn Treebank that were released in 2003 (reconciliation of the ISIP Switchboard segmentations and transcriptions with the Penn Treebank segmentations and transcriptions). We introduced the mainstream community to advanced statistical modeling techniques such as Support Vector Machines and enhanced these for NLP applications.

Further, in line with the primary goal of the ITR program, this project created close collaborations between groups who did not previously work together. The PIs collaborated on a number of new initiatives as offshoots of this project, including applications in parsing, information retrieval, and homeland security. A subset of the PIs participated in conversational speech evaluations and workshops (e.g., DARPA EARS). Hence, we can conclude that this project created new synergies and new research directions that will continue beyond the timeframe of this project.

In this final report, we briefly describe some of the significant findings of our research below.

A. Laboratory for Linguistic Information Processing, Brown University

Learning general functional dependencies, i.e. functions between arbitrary input and output spaces, is one of the main goals in supervised machine learning. Recent progress has to a large extent focused on designing flexible and powerful input representations, for instance by using kernel-based methods such as Support Vector Machines. We have addressed the complementary issue of problems involving complex outputs such as multiple dependent output variables and structured output spaces. In the context of this project we have mainly dealt with the problem of label sequence learning, a class of problems where dependencies between labels take the form of nearest neighbor dependencies along a chain or sequence of labels. The latter is a natural generalization of categorization or multiclass-classification that has many applications in the context of natural language processing and information extraction. Special cases include part-of-speech tagging, named entity recognition, and speech-accent prediction. More specifically, we have developed and empirically investigated several extensions of state-of-the-art categorization algorithms such as AdaBoost, Support Vector Machines, and Gaussian Process classification. We have designed and implemented several scalable learning algorithms that combine standard optimization techniques employed in the context of the above mentioned methods with dynamic programming techniques that account for the nearest neighbor dependencies. Experimental evaluations on a wide variety of tasks have shown the competitiveness of these methods compared to existing techniques like Hidden Markov Models and Conditional Random Fields.

A second line of research conducted in the context of the present ITR has dealt with ways to systematically exploit class hierarchies and taxonomies. The main question we have investigated is

whether or not a priori knowledge about the relationships between classes helps in improving classification accuracy, in particular in cases with many classes and few training examples. This is highly relevant for applications like word sense disambiguation and text categorization, where the number of classes can easily be in the tens of thousands. To that extend we have focused on a hierarchical version of the well-known perceptron learning algorithm as well as an extension of multiclass Support Vector Machines. We have shown that this approach can be effective in situations with sparse training data.

B. Center for Language and Speech Processing, Johns Hopkins University

The Structured Language Model (SLM) aims at making a prediction of the next word in a given word string by making a syntactical analysis of the preceding words. However, it faces the data sparseness problem because of the large dimensionality and diversity of the information available in the syntactic parses. A neural network model is better suited to tackle the data sparseness problem and its use has been shown to give significant improvements in perplexity and word error rate over the baseline SLM (Emami et al, 2003).

In this work we have investigated a new method of training the neural net based SLM. Our model makes use of a neural network for that component of the SLM that is responsible for predicting the next work given the previous words and their partial syntactic structure. We have investigated both a mismatched and a matched training scenario. In matched training, the neural network is trained on partial parses similar to those that are likely to be encountered during evaluation. On the other hand in the mismatched scenario, faster training time is achieved but at the cost of mismatch between training and evaluation and hence, possible degradation in performance.

The Structured Language Model works by assigning a probability P(W,T) to every sentence W and every possible binary parse T of W. The joint probability P(W,T) of a word sequence W and a complete parse T is broken into:

$$P(W,T) = \prod_{k=1}^{n+1} P(W_k \mid W_{k-1}T_{k-1}) \cdot P(t_k \mid W_{k-1}T_{k-1}, W_k) \cdot \prod_{i=1}^{N_k} P(p_i^k \mid W_{k-1}T_{k-1}, w_k, t_k, p_1^k \cdots p_{i-1}^k)$$

where $W_{k-1}T_{k-1}$ is the word-parse (k-1)-prefix, t_k is the tag assigned to w_k by the TAGGER, $N_k - 1$ is the number of operations the CONSTRUCTOR executes at sentence position k before passing control to the PREDICTOR, and p_i^k denotes the i-th CONSTRUCTOR operation carried out at position k in the word string.

Subsequently, the *language model* probability assignment for the word at position k+1 in the input sentence is made using:

$$\mathrm{P}_{\mathrm{SLM}}(w_{k+1} \mid W_k) = \sum_{T_k \in S_k} P(w_{k+1} \mid W_k T_k) \cdot \rho(W_k T_k)$$

$$\rho(W_k T_k) = P(W_k T_k) \Big/ \sum_{T_k \in S_k} P(W_k T_k)$$

which ensures a proper probability normalization over strings W^* where S_k is the set of all parses built and retained by the model at the current stage k.

Neural networks are very suitable for modeling conditional discrete distribution with large vocabularies. These models work by first assigning a continuous feature vector with every token in the vocabulary, and then using a standard multi-layered neural net to get the conditional distribution at the output, given the input feature vectors. Training is achieved by searching for parameters Θ of the neural network and the values of feature vectors that maximize the penalized log-likelihood of the training corpus:

$$L = \frac{1}{N}\sum_{t} \log p(y^t \mid x_1^t, x_2^t, \cdots, x_{n-1}^t; \Theta) - R(\Theta)$$

where $p(y^t \mid x_1^t, x_2^t, \cdots, x_{n-1}^t; \Theta)$ is the probability of word y^t (network output at time t), N is the training data size and R(Θ) is a regularization term, L-2 norm squared of the parameters in our case.

We have used a neural net to model the SCORER component of the SLM. By the SCORER we refer to the model $P(w_{k+1} \mid W_k T_k)$. The neural net SCORER's parameters can be obtained by training it on the events extracted from the gold standard (usually one best) parses obtained from an external source (humans or an automatic parser). However, there would be a mismatch during evaluation since the partial parses during that phase are not provided and have to be hypothesized by the SLM itself. We have called the SCORER trained in this manner the *mismatched* SCORER.

On the other hand, one can train the model on partial parses hypothesized by the baseline SLM, thus maximizing the proper log-likelihood function. We have called this procedure the *matched* training of the SCORER.

Experimental results have shown considerable improvement in both perplexity and WER when using a neural net based SLM, specially in the case of matched SCORER training. On the UPenn section of the WSJ corpus, perplexity reductions of 12% and 19% over the baseline SLM (with a perplexity of 132) have been observed when using the mismatched and matched neural net models respectively.

For the WER experiments, the neural net bases models were used to re-rank an N-best list output by a speech recognizer on the WSJ DARPA'93 HUB1 test set (with a 1-best WER of 13.7%). The mismatched and matched neural net models reduced the SLM baseline WER of 12.6% to 12.0% and 11.8% (for relative improvements of 4.8% and 6.3%) respectively.

In summary, neural network models showed to be capable of taking advantage of the richer probabilistic dependencies extracted through syntactic analysis. In our case the use of a neural net for the SCORER component of the Structured Language Model resulted in considerable improvements in both perplexity and Word Error Rate (WER) with the best results achieved when using a training procedure matched with the evaluation.

C. Signal, Speech, and Language Interpretation Lab, University of Washington

Prosody can be thought of as the "punctuation" in spoken language, particularly the indicators of phrasing and emphasis in speech. Most theories of prosody have a symbolic (phonological) representation for these events, but a relatively flat structure. In English, for example, two levels of prosodic phrases are usually distinguished: intermediate (minor) and intonational (major) phrases [Beckman & Pierrehumbert, 1986]. While there is evidence that both phrase-level emphasis (or, prominence) of words and prosodic phrases (perceived groupings of words) provide information for syntactic disambiguation [Price et al., 1991], the most important of these cues seems to be the prosodic phrases or the boundary events marking them. While prior work has looked at the use of prosody in automatic parsing of isolated sentences, a key component of our work involved sentence detection as well, since our goal is to handle continuous conversational speech. Hence, the focus of our work has been on automatically recognizing sentence boundaries and sentence-internal prosodic phrase structure and investigating methods for integrating that structure in parsing.

To support these efforts, we also worked on analysis of acoustic cues to prosodic structure. The most important (and best understood) acoustic correlates of prosody are continuous-valued, including fundamental frequency (F0), energy, and duration cues. Particularly at phrase boundaries, cues include significant falls or rises in fundamental frequency accompanied by word-final duration lengthening, energy drops, and optionally a silent pause. In addition, however, there is evidence of spectral cues to prosodic events, so some of our work explored these cues, which also have implications for improving speech recognition.

Our approach to integrating prosody in parsing is to use symbolic boundary events that have categorical perceptual differences, including prosodic break labels that build on linguistic notions of intonational phrases and hesitation phenomena but also higher level structure. These events are predicted from a combination of the continuous acoustic features, rather than using the acoustic features directly, because the intermediate representation simplifies training with high-level (sparse) structures. Just as most automatic speech recognition (ASR) systems use a small inventory of phones as an intermediate level between words and acoustic feature to have robust word models (especially for unseen words), the small set of word boundary events are also useful as a mechanism for generalizing the large number of continuous-valued acoustic features to different parse structures. This approach is currently somewhat controversial because of the high cost of hand labeling, and to some extent because of its association with a particular linguistic theory. However, the specific subset of labels used in this work are relatively theory neutral and language independent, and a key contribution of this work is the use of weakly supervised learning to reduce the cost of prosodic labeling.

An alternative approach, as in [Noth et al, 2000], is to assign categorical "prosodic" labels defined in terms of syntactic structure and presence of a pause, without reference to human perception, and automatically learn the association of other prosodic cues with these labels. While this approach has been very successful in parsing speech from human-computer dialogs, we expect that it will be problematic for conversational speech because of the longer utterance and potential confusion between fluent and disfluent pauses.

C.1 Data, Annotation and Development

The usefulness of speech databases for statistical analysis is substantially increased when the database is labeled for a range of linguistic phenomena, providing the opportunity to improve our understanding of the factors governing systematic suprasegmental and segmental variation in word forms. Prosodic labels and phonetic alignments are available for some read speech corpora, but only a few limited samples of spontaneous conversational speech have been prosodically labeled. To fill this gap, a subset of the

Switchboard corpus of spontaneous telephone-quality dialogs was labeled using a simplification of the ToBI system for transcribing pitch accents, boundary tones and prosodic constituent structure [Pitrelli et al., 1994]. The aim was to cover phenomena that would be most useful for automatic transcription and linguistic analysis of conversational speech. This effort was initiated under another research grant, but completed with partial support from this NSF ITR grant.

The corpus is based on about 4.7 hours of hand-labeled conversational speech (excluding silence and noise) from 63 conversations of the Switchboard corpus and 1 conversation from the CallHome corpus. The orthographic transcriptions are time-aligned to the waveform using segmentations available from Mississippi State University (http://www.cavs.msstate.edu/hse/ies/projects/switchboard/index.html) and a large vocabulary speech recognition system developed at the JHU Center for Language and Speech Processing for the 1997 Language Engineering Summer Workshop (www.clsp.jhu.edu/ws97) [Byrne et al., 1997]. All conversations were analyzed using a high quality pitch tracker [Talkin, 1995] to obtain F0 contours, then post-processed to eliminate errors due to crosstalk. The prosody transcription system included: i) breaks (0-4), which indicate the depth of the boundary after each word; ii) phrase prominence (none, *, *?); and iii) tones, which indicate syllable prominence and tonal boundary markers. It also provides a way to deal with the disfluencies that are common in spontaneous conversational speech (a p diacritic associated with the prosodic break), and to indicate labeler uncertainty about a particular transcription. The annotation does not include accent tone type, primarily to reduce transcription costs and because we hypothesized that the phrase tones would be relevant to dialog act representations which may be relevant for question-answering. For further information on the corpus and an initial distributional analysis, see [Ostendorf et al. 2001].

The prosodically labeled subset of Switchboard overlaps with the subset of that corpus annotated with Treebank parses, but there is a mismatch in the orthographic transcriptions because the Treebank parses were based on an earlier version of transcripts and the prosodic annotation was based on the higher quality corrections done by Prof. Picone's group (ISIP) at Mississippi State. In addition, we made use of the DARPA EARS metadata annotations that overlapped with the Treebank parses, which where again based on the higher quality transcriptions. To be able to use all of these resources, we used an alignment of words provided by the ISIP team, and mapped the Treebank parse information to the more recent word transcriptions, which could then be aligned with the EARS metadata annotations. Differences in transcriptions were handled by: dropping the parse information for deletions, transferring it as is for word substitutions, and treating it as "missing" information for insertions in the corrected transcripts. While most of the differences between the Treebank and corrected word transcriptions involved simple substitutions (or deletions) that had little or no impact on the parse (e.g. "a" vs. "the"), there were some cases where the transfer introduced noise into the collection of parses. The most frequent such cases were in disfluent regions, where transcribers tend to have more difficulties, including missed word fragments or repetitions ("I I" vs. "I I I"). An additional difference between the Treebank parses and the EARS metadata annotations is the marking of sentence boundaries. Since speakers frequently begin sentences with conjunctions, the metadata conventions often split up constituents marked as compound sentences in Treebank. Because the metadata labelers listened to the speech and the Treebank labelers did not, we chose to use the metadata constituents, which in most cases involved simply dropping a top-level (S) node, but in some cases involved adding a top-level node called "SUGROUP".

C.2 Automatic Labeling of Prosodic Structure

An important part of the effort was development of an automatic prosodic labeling system that would provide cues to improve parsing. In addition, the resulting system was inspected to analyze possible dependencies between prosodic and parse structures in conversational speech. In the experiments, we used decision tree classifiers with different combinations of acoustic, punctuation, parse, and disfluency cues. While more sophisticated techniques, such as HMMs and maximum entropy models, have been

used for related tasks of sentence boundary detection (see [Liu et al., 2005] for a brief survey), we chose decision trees because they are easy to inspect for learning about the prosody-syntax relationship and because this simplified the weakly supervised learning experiments, which were the focus of our efforts.

For the prosody/syntax analyses, we designed trees to predict prosodic labels from syntactic structure, as well as trees to predict prosodic structure from a combination of syntactic and acoustic cues. For purposes of providing information to a parser, we designed trees to predict prosodic constituents from acoustic cues and part-of-speech (POS) tags, but as an intermediate step in designing these trees we also used syntactic cues in designing trees as part of the weakly supervised training. More specifically, a small set of labeled data was used to train prosody models based on both text and acoustic cues, which were then used in combination to automatically label a large set of data that had not been hand-annotated with prosodic structure, and finally new (separate) acoustic-based prosody models were designed from this larger data set for use in parsing new data.

Experiments were conducted on the Switchboard corpus, using the prosodically annotated subset described above for initial training and evaluation (independent subsets for each). Then the full Switchboard training set was incorporated using various methods for weakly supervised learning, as described below. The prosodic constituent labels were merged into 3 classes: major intonational phrase boundary (4), hesitation boundary (1p, 2p), and all other fluent word boundaries. We grouped minor intonational phrase boundaries (3) with the default word boundary class, because preliminary experiments showed that they were almost never predicted by the decision trees (even with sampled training to account for the low frequency) and because they were most often confused with the default word class in 4-class prediction experiments. The simple 3-class system also has the advantage that it is relatively theory neutral and language independent in that essentially all languages have a notion of fluent and disfluent segmentation.

The acoustic cues included normalized F0, energy and duration cues based on those used in [Kim et al., 2004] and similar to those used in other metadata detection studies [Shriberg et al., 2000]. Text-based cues -- including punctuation, parse structure and disfluency markers -- were taken from the annotations available from the Linguistic Data Consortium, aligned to the word transcriptions as described above. The parse features included right-to-left and left-to-right relative depth, part-of-speech, label of the left and right syntactic constituents, and parenthetical markers. In addition, disfluency interruption points and flags for filled pauses and sentence-initial conjunctions were used as features. Punctuation as inserted by a human transcriber (including incomplete sentences) and estimated speaker turn boundaries (defined simply as a word boundary with a silence of length greater than 4s) were also used.

The baseline system trained on hand-labeled data has error rates of 9.6% when all available cues are used (both syntax and prosody) and 16.7% when just acoustic and POS cues are used. This can be compared to an error rate of 30% when the default class is assigned to all word boundaries. We considered three different weakly supervised training techniques for adding data without prosodic labels (but with hand-labeled syntactic structure) into the training set: EM, co-training, and self-training. The co-training algorithm used classifiers designed on either acoustic or syntactic cues, and it differed slightly from the standard method in that we used an information-theoretic distance on the tree posteriors to determine when to omit samples with conflicting classifier decisions. The self-training algorithm used bagging with uniform class sampling to deal with data skew [Liu et al., 2004]. In all cases, only 1-2 iterations were needed. Both the co-training and EM approaches gave improved performance over the baseline, with the

EM algorithm giving the best results of 14.2% error for the acoustic-only trees, which corresponds to a 15% reduction in error rate over supervised training. The self-training strategy actually hurt performance.[1]

From analysis of the resulting trees, we find that punctuation and repair points are correlated with prosodic breaks and hesitations, as expected. Silence duration is the most useful individual acoustic feature, but alone it is not a reliable predictor of prosodic phrases, since there are many prosodic phrases that do not occur at silences and because silences are frequently associated with hesitations. Aside from syntactic structure, the most important text features for predicting prosodic constituents are punctuation, disfluency edit point markers, filler words (sentence-initial coordinating conjunctions, discourse markers, filled pauses), and turn boundaries. Some important syntactic features include depth of subtrees on the left and right sides of the boundary, previous and next syntactic constituent tag, length of closing phrase, and part-of-speech tags. These features were relevant when associated with the target word boundary, but frequently also with the next or previous word boundary. Surprisingly, the label of the joining constituent is not useful. This analysis provided input into the parse reranking work described in the next section.

Due to the success of the weakly supervised training on prosodic phrase boundary detection, we have recently started investigating use of the same technique for training models of prosodic prominence. Initial results show only a 4% reduction in error rate for the system based on acoustic cues, from 22% to 21% error. Despite the high error rate, however, the automatically annotated prominence appears to be useful in topic identifications in preliminary experiments associated with a separate NSF project (IIS-0121396).

C.3 Prosody and Parsing

Parsing speech can be useful for a number of tasks, including information extraction and question answering from audio transcripts. However, parsing conversational speech presents a different set of challenges than parsing text: sentence boundaries are not well-defined, punctuation is absent, and presence of disfluencies (edits and restarts) impact the structure of language. Most prior work on parsing conversational speech has focused on handling disfluencies [Hindle 1983; Mayfield 1995; Charniak & Johnson, 2001], but experiments relied on hand-marked sentence boundaries and made use of punctuation as in text-based parsers. While utterance-level segmentation may be reasonable to assume in current human-computer dialog systems, it is not realistically available in recognized conversational speech. Hence, our work looked at the problem of parsing text with disfluencies and without punctuation.

We have investigated three main issues in the use of prosody in parsing: the impact of automatic sentence segmentation, the usefulness of interruption points, and the usefulness of automatically detected sub-sentence prosodic constituent boundaries (described above). In all cases, we use a two-stage architecture where metadata (constituent boundaries) are first detected with a combination of prosodic and simple text features, and then these symbolic events (or their posterior probabilities) are used in parsing. Our approach focuses on categorical boundary events, which are predicted from a combination of acoustic features, rather than using the acoustic features directly. As argued earlier, the intermediate representation simplifies training with sparse structures. Key research issues include whether the metadata should be treated as "words" or as features on words, whether edits should be represented with an independent component, and how to represent uncertainty of the metadata classifiers. Our work has begun investigating all of these questions, but some remain unanswered and are being pursued in ongoing work.

---

[1] The results reported here are in some cases worse than those reported in an earlier progress report, because they are based on a larger data set. Due to a data processing bug, several files were omitted from earlier studies. In addition, because of the larger amount of data used and the richer feature sets, the trees are much larger than those described in prior reports.

The data used in this work is the Treebank portion of the Switchboard corpus of conversational telephone speech, which includes sentence-like unit boundaries (SUs) as well as the reparandum and interruption point of disfluencies. The data consists of 816 hand-transcribed conversation sides (566K words), of which we reserve 128 conversation sides (61K words) for evaluation testing according to the 1993 NIST evaluation choices. In all cases, training was based on hand-labeled SUs. Parses were evaluated using SU boundaries rather than the standard punctuation-based units that the Treebank is based on, so the gold standard parses and parse evaluation metric were modified to incorporate the SUs.

The most exhaustive series of experiments looked at the impact of automatic segmentation on parsing. For this particular effort, we chose to work with the complete word sequence, i.e. including all of the words within edit regions, to allow experimentation with multiple parsers. In initial work [Kahn et al., 2004], we used the structured language model (SLM) as a parser with a simple pause-based segmentation and automatically detected SUs (69% vs. 35% slot error rate, respectively), showing a significant improvement in parsing performance when using the automatic SUs. We then confirmed the findings with results on the same segmentation alternatives but using two state-of-the-art statistical parsers trained on conversational speech: the Bikel [Bikel, 2004] [2] and Charniak [Charniak & Johnson, 2001] [3] parsers. Figure 1 shows the relative performance of each parser for the two different segmentations, comparing to the oracle performance for each using hand-annotated SUs. In order to have a single number for performance, we use the F-measure calculated from bracket precision and recall. (Trends with separate precision and recall measures and with bracket crossing statistics follow the same patterns.) For all three parsers, there is a significant gain in performance due to using the explicit SU detection algorithm, with more than half of the performance loss associated with the pause-based segmenter recovered when moving to the more sophisticated SU detection system. As SU detection improves, we would expect further performance gains. Also important is the observation that the impact of increased SU error is not lessened by using a better parser: the best oracle results were achieved with the Bikel parser, but it was much more sensitive to SU errors than the SLM.

In trying to understand the role of IPs in parsing, we ran several experiments, including some with and without human-annotated punctuation. Without punctuation, there was a small increase in parsing performance of the SLM using IPs When the SUs are automatically detected. We were not able to confirm these gains with other parsers; however, recent work in [Johnson, Charniak & Lease, 2004] shows a benefit to edit detection from using IPs which presumably would lead to improved parsing in their two-stage processing strategy [Johnson & Charniak, 2004]. Including punctuation and IPs in experiments with the SLM showed an

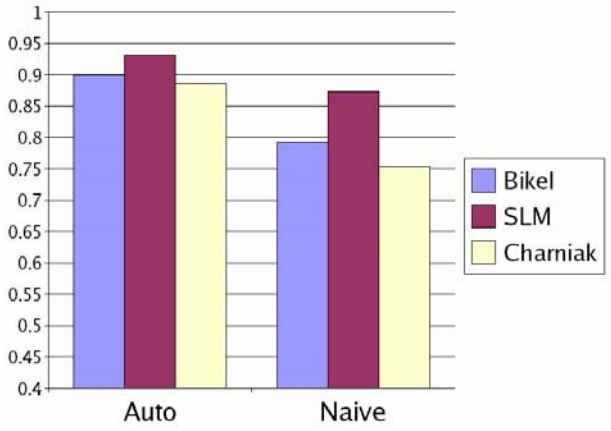

Figure 1. Relative performance of different parsers for the RT03 MDE SU detection system and a simple pause-based system, in each case comparing the F-measure to that obtained with oracle SUs.

[2] http://www.cis.upenn.edu/~dbikel/download.html (Version 0.9.9). For this work, we trained the Bikel parser on the Switchboard Treebank parses with the Collins settings.

[3] ftp://ftp.cs.brown.edu/pub/nlparser/ (August 2004)

interesting trade-off of bracket precision and recall using the two different cues, as illustrated in Figure 2. We saw the improved precision associated with using both punctuation and IPs as possible evidence that sub-sentence prosodic constituents might be useful.

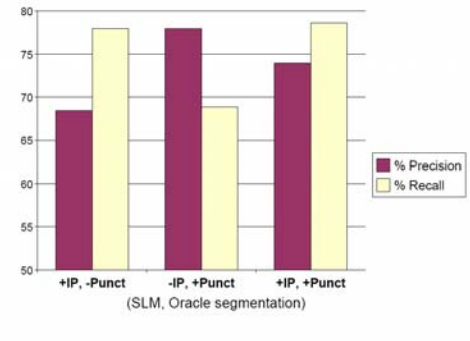

Figure 2. Precision-recall trade-offs associated with using punctuation, oracle IPs or both in addition to SUs in parsing.

In all of the above work, metadata is incorporated as "word" tokens, similar to the standard mechanism for parsers to incorporate punctuation. For sentence segmentation with a reasonably reliable segmenter, this may make sense, but certainly for sub-sentence prosodic constituents there is the potential for the gains associated with adding prosody to be offset by a loss from the extra words blocking part of the history that might be used in a statistical model of word dependence. We conjecture that this may in part explain the negative results obtained in [Gregory et al., 2004], since our analyses of the prosody prediction trees provides some evidence that sub-sentence prosodic constituents may be useful in parsing. (The direct use of acoustic features may also be problematic.) In addition, the use of metadata events as "words" requires a hard decision in the first stage of detection, and many results in speech processing suggest that soft decisions (e.g. using class posteriors) are more effective.

To address these problems, we developed an extension to the SLM that uses prosodic constituents as hidden conditioning variables, similar to headword conditioning in the SLM. However, since our subsequent work obtained much better baseline performance with other parsers, we decided to explore a parse reranking framework [Johnson et al., ms. in prep.] as an alternative method for incorporating automatically detected prosodic constituents. The approach uses a maximum entropy reranking model and introduces new features based on counts of syntactic constituent types weighted by the posterior probability of different prosodic events. Experiments with this new approach are in progress, now under other funding, and we anticipate having results in early 2005. This series of experiments will also look at the question of whether a separate stage of edit detection benefits parsing compared to simply incorporating the edit structure in the parser with the same status as other constituents.

C.4 PROSODY AND ACOUSTIC MODELING

Most research on the use of prosody in automatic speech processing has focused on F0, energy and duration correlates to prosodic structure. However, there is evidence from long standing acoustic, articulatory and perceptual studies of speech suggesting that there are spectral correlates as well. For that reason, we conducted an analysis of our prosodically labeled conversational speech data using acoustic parameters and clustering techniques that are standard in speech recognition. We found that prosodic factors are associated with acoustic differences that can be learned in standard speech recognition systems. Both prosodic phrase structure and phrasal prominence seem to provide distinguishing cues, with some phones being affected much more than others (as one would expect from the linguistics literature). We hypothesized that we would find that constituent onsets were important at all levels (syllable, word and prosodic phrase). Instead, we found that onset is more important for syllables, but constituent-final position is more important at higher levels. Prosodic prominence had a smaller affect than phrase structure in terms of increasing likelihood of the training data, but seemed to result in more separable models when it did play a role.

Finally, we found evidence that segmental cues can help distinguish fluent from disfluent phrase boundaries, in that segments associated with these categories are frequently placed in different clusters. These differences can be leveraged in a "multiple pronunciation" acoustic model to aid in detecting fluent vs. disfluent prosodic boundaries, though additional prosodic cues are necessary to separate these from unmarked word boundaries. A limitation of this work was that it was based on hand-labeled data, and therefore did not take advantage of the full training data set needed for designing a state-of-the-art recognition system. However, with our recent developments in prosodic annotation, we will be able to assess the usefulness on a much larger corpus in the future.

D. Institute for Signal and Information Processing, Mississippi State University

Hidden Markov models (HMMs) with Gaussian emission densities are the prominent modeling technique in speech recognition. HMMs suffer from an inability to learn discriminative information and are prone to overfitting and overparameterization. Recent work in machine learning has focused on models, such as the support vector machine (SVM), that automatically control generalization and parameterization as part of the overall optimization process. SVMs, however, require ad hoc (and unreliable) methods to couple it to probabilistic speech recognition systems. We have applied a probabilistic Bayesian learning machine termed the relevance vector machine (RVM) as the core statistical modeling unit in a speech recognizer. The RVM is shown to provide superior performance compared to HMMs and SVMs in terms of both accuracy and sparsity on a continuous alphadigit task.

Computer speech recognition is typically posed as a statistical pattern recognition problem based on Bayesian methods in which the acoustic model and language model are treated as separate statistical models. The focus of our work has been the acoustic model, which maps sequences of features vectors to probabilities that these vectors were produced by a given linguistic unit, such as phone. In most state of the art recognition systems, a hidden Markov model (HMM) is used as the acoustic model. The popularity of the HMM representation is based on an HMMs ability to simultaneously model the temporal progression of speech (speech is usually seen as a "left-to-right" process) and the acoustic variability of the speech observations. The temporal variation is modeled via an underlying Markov process while the acoustic variability is modeled by an emission distribution at each state in the Markov chain.

The most commonly used emission distribution is the Gaussian mixture model (GMM). While combinations of HMMs and Gaussian mixture models have been extremely successful, there are two fundamental limitations of these approaches: (1) the parametric form of the underlying distribution is assumed to be Gaussian, (2) the maximum likelihood (ML) approach, which is typically used to estimate model parameters, does not explicitly improve the discriminative ability of the model. The ML approach maximizes the probability of the correct model while implicitly ignoring the probability of the incorrect model. Ideally, a training approach should force the model toward in-class training examples while simultaneously driving the model away from out-of-class training examples. Methods such as maximum mutual information and minimum classification error have been developed to incorporate discriminative training directly into the standard HMM/GMM framework. However, their success has been limited due primarily to their considerable computational costs.

The weaknesses of the HMM/GMM system have led researchers to explore other models, such as hybrid connectionist systems, which merge the power of artificial neural networks (ANNs) and HMMs. HMM/ANN hybrids have shown promise in terms of performance but have not yet found widespread use due to some serious problems. First, ANNs are prone to overfitting the training data if allowed to blindly converge. To avoid overfitting, a cross-validation set is often used to define a stopping point for training. This is wasteful of data and resources — a serious consideration in speech where the amount of labeled training data is limited. ANNs also typically converge much slower than HMMs. Most importantly, the

HMM/ANN hybrid systems have not shown the substantial improvements in recognition accuracy over HMM/GMM systems that would be necessary to force a paradigm shift.

The approaches developed in this work draw significantly from the HMM/ANN hybrid systems. However, we seek methods which are automatically immune to overfitting without the artificial imposition of a cross-validation set as well as methods which can automatically learn the appropriate model structure as part of the overall optimization process. One such model that has come to the forefront of pattern recognition research is the support vector machine (SVM). The support vector paradigm is based upon structural risk minimization (SRM) in which the learning process is posed as one of optimizing some risk function. The optimal learning machine is the one whose free parameters are set such that the risk is minimized.

Finding a minimum of the risk function is typically impossible due to the unknown distribution. Instead, it has been shown that a relationship exists between the actual risk, which is related to the empirical risk (i.e. the training set error which can be measured) and the Vapnik-Chervonenkis (VC) dimension. The VC dimension is a measure of the capacity of a learning machine to learn any training set and is typically closely related to the complexity of the learning machine's structure. With this result, we can guarantee both a small empirical risk (training error) and good generalization — an ideal situation for a learning machine.

In their most basic form SVMs use the SRM principle to impose an order on the optimization process by ranking candidate separating hyperplanes based on the margin they induce. For separable data, the optimal linear hyperplane is the one that maximizes the margin. The true power of the SVM, however, lies in how it deals with nonlinear class separating surfaces. Providing for a nonlinear decision region is accomplished using kernels. The optimization process yields a decision function where the sign of can be used to classify examples as either in-class or out-of-class. The decision function is formed from only those training vectors that lie on the margin or in overlap regions. In practice, the proportion of the training set that becomes support vectors is small, making the classifier sparse. Consequently, the training process, along with the training set, directly optimize the complexity of the learning machine. In contrast, ANN systems often make *a priori* assumptions about the form of the model.

SVMs have had great success on static classification tasks. However, it is only recently, that these techniques have been applied to continuous speech recognition. While the SVMs provide an excellent classification paradigm, they suffer from two serious drawbacks that hamper their effectiveness in speech recognition. First, while sparse, the size of the SVM models (number of non-zero weights) tends to scale linearly with the quantity of training data. For a large speaker independent corpus this effect is prohibitive. Second, the SVMs are binary classifiers. In speech recognition this is an important disadvantage since there is significant overlap in the feature space which can not be modeled by a yes/no decision boundary. Further, the combination of disparate knowledge sources (such as linguistic models, pronunciation models, acoustic models, etc.) requires a method for combining the scores produced by each model so that alternate hypotheses can be compared. Thus, we require a probabilistic classification which reflects the amount of uncertainty in our predictions.

We have investigated a Bayesian model termed the relevance vector machine (RVM) which is similar in form to the SVM but which addresses these two problems. The essence of an RVM is a fully probabilistic model with an automatic relevance determination prior over each model parameter. Thus, sparseness in the RVM model is explicitly sought in a probabilistic framework.

D.1 Sparse Bayesian Methods

Supervised learning in speech recognition implemented via a maximum likelihood approach is the dominant approach for finding values of the parameters in our model that best match the training data. Our expectation in data modeling is that given sufficient training data, the model would generalize to unseen test sets. Two levels of inference must be implemented to accomplish this. First, assuming that a particular model is true, we seek to infer the values for the parameters of the model that best fit the data at hand. This is exactly the training process used in the ANN hybrids. Second, we must decide which model is most appropriate given the data at hand, i.e. model comparison.

A simple approach to model comparison might dictate that we simply choose the model that fits the data best — the maximum likelihood solution. However, a more complex model can always fit the data better. Jaynes describes an extreme interpretation of this problem where we would always choose the so-called *Sure Thing* hypothesis, under which exactly the training set and only the training set is possible. Though it is the maximum likelihood solution, the Sure Thing hypothesis is intuitively displeasing and is counter to our desire for a solution which generalizes. We avoid choosing the Sure Thing hypothesis by expressing an *a priori* preference for simpler solutions using the principle of Occam's Razor. MacKay and others have formalized this preference mechanism through the use of Bayesian methods. These provide a natural and quantitative embodiment of Occam's razor. The first level of inference requires that we find the best-fit parameters. The second level of inference requires the comparison of competing hypotheses. If we assume that the competing hypotheses are *a priori* equiprobable then the best hypothesis is chosen by evaluating the evidence. The evidence is computed by marginalization across the model parameters.

The evidence provides a natural trade-off between the best-fit likelihood and the Occam factor. This concept is closely related to other methods such as the Minimum Description Length and the Bayesian Information Criteria where the model is directly penalized by the number of parameters used. A similar idea was also incorporated into SVM models, which penalize the models with too large a capacity (VC dimension). However, while the SVM models are forced to estimate the penalty via cross-validation schemes, Bayesian techniques automatically determine and apply the penalty in a fully probabilistic framework.

Assuming we have no prior knowledge that would cause us to favor a particular prior, we can find the optimal value for by evaluating the evidence. If we did have prior knowledge, we would simply repeat the inference over using the prior. At some level of the inference, we will arrive at a point where our prior knowledge is too weak to apply and then we evaluate the evidence. This extreme application of the Bayesian prior as a control parameter is known as the method of *automatic relevance determination* (ARD). It is so named because the prior over the input unit weights in a neural network can 'shut-off' those input dimensions which are irrelevant to the problem at hand. ARD is at the heart of the relevance vector machines that will be described next.

D.2 Support Vector Machine

SVM is powerful tool for distinguishing each class with a nonlinear system basis function. The fundamental idea of an SVM is to project the input space vectors to a high-dimensional feature space using a nonlinear map, which is defined as kernel function [Wan et al., 2003]. The Structural Risk Minimization (SRM) principle enables one to implement the SVM, since the SRM defines the boundary for training model errors and confidence interval via the VC dimension. The hyperplane will separate the class depending on the binary or n-class cases while SVM reduces the empirical risk. The power of SVMs lies in their ability to transform data to a higher dimensional space and construct a linear binary classifier in the higher dimensional space [Ganapathiraju et al., 2002]. A linear hyperplane in the higher dimensional space transforms to a complex nonlinear decision region in the input feature space.

For improving the efficiency and performance of SVM, the score-space kernel has been investigated for computation and performance. Wan [5] has proposed the score-space kernel method that simulates the human disriminiation process. In terms of speaker recognition system, humans distinguish identity using the intonation, accent, and frequently occuring words. Since the speech utterance for one speaker is highly correlated between segments of speech, taking the score of whole utterance as factor will improve the system. The following sections briefly explain the score-space approach.

D.2.1 Generative Kernel Function

The kernel function is derived from a generative probability model [Jaakkola et al., 1999]. Even though discriminative methods are proved to be superior to the generative models for classification problems, generative methods are excellent for extracting information from input features. Kernel methods are suitable for using discriminative classification with a generative probability model. Suppose, the training set is composed of X_i and the corresponding binary targets are Y_i. The targets of new training examples are obtained from a weighted sum of the training targets. The estimated targets are consisted of estimating of weights and kernel functions. The weights represent the overall importance of the each training example X_i, and the kernel function compute the closeness of the pair of datasets. The estimated target process is represented by:

$$\hat{Y} = sign(\sum_i Y_i w_i K(X_i, X)) \tag{1}$$

The kernel function should be derived by a probabilistic method. The probabilistic method measures the difference between input sample X_i and test sample X. The training targets can be assumed to have a logistic regression distribution, and the targets are estimated given input data X and parameter vector θ.

$$P(Y \mid X, \theta) = \sigma(Y\theta^T X) \quad \text{where } \sigma(y) = (1 + e^{-y})^{-1} \tag{2}$$

By assigning a distribution for θ like a zero mean Gaussian with a full covariance matrix ∑, the posterior distribution for training targets can reduce the complexity of model. The maximum a posteriori (MAP) estimate for the parameters θ given a training set of examples is found by maximizing the following penalized log likelihood:

$$\sum_i \log P(Y_i \mid X_i, \theta) + \log P(\theta) = \sum_i \log \sigma(Y_i \theta^T X_i) - \frac{1}{2}\theta^T \Sigma^{-1} \theta + c \tag{3}$$

where the constant c does not depend on θ. The similarity between data X_i and X can be captured by taking the gradient space of the model. The gradient of the generative model with respect to a parameter describes how that parameter contributes to the process of generating a particular data set. The posterior distribution over training targets are finally estimated by

$$P(Y \mid X, \theta) = \sigma(Y \sum_i Y_i w_i (X_i^T \Sigma X)) \tag{4}$$

Comparing the equation (1) and (4), the kernel function can de replaced by $K(X_i, X) = X_i^T \Sigma X$. Through these processes the generative properties are involved in kernel functions, and SVM is able to exploit the generative and discriminative properties at the same time.

D.2.2 Score-Space Kernels

We investigated a more specific kernel function which enables us to classify variable length sequences of input vectors in a space of fixed dimension, called the score-space [Wan et al., 2003]. The score-space kernel uses any parametric generative model to classify whole sequences. The space to which sequences are mapped is called the score-space, so named because it is defined by and derived from the likelihood score, p(X|M,θ) of a generative model M. Given a set of k generative models the generic formulation of the mapping of a sequence, X={$\mathbf{x}_1$, …,$\mathbf{x}_{Nl}$ }, to the score-space is

$$\Psi_{\hat{F}}^{f}(X) = \Psi_{\hat{F}} f(\{p_k(X \mid M_k, \theta_k)\}). \tag{5}$$

This equation consists of score-argument $f(\{p_k(X \mid \theta_k)\})$, which is a function of scores of a set of generative model, and score-mapping operator $\Psi_{\hat{F}}$, which maps the scalar score-argument to the score-space. Any function may be used as a score-argument. We deal with two specific cases that lead to the likelihood score-space kernel and the likelihood ratio score-space kernel. By setting the score-argument to be the log likelihood of a single generative model, M, parameterized by θ, and choosing the first derivative score-operator, we obtain the mapping for the likelihood score space.

$$\Psi(X) = \nabla_\theta \log P(X \mid M, \theta) \tag{6}$$

Each component of the score-space, $\Psi(X)$, corresponds to the derivative of the log likelihood score with respect to one of the parameters of the model. This mapping is known as the Fisher mapping. The gradient of the log likelihood with respect to a parameter describes how that parameter contributes to the process of generating a particular speaker model. For the exponential family of distributions, these gradients form sufficient statistics for the models. This gradient space also naturally preserves all the structural assumptions that the model encodes about the generation process. When the gradients are small then likelihood has reached a local maximum and vice versa.

Using the first derivative with argument score-operator and the same score-argument the mapping becomes

$$\Psi(X) = \begin{bmatrix} \log P(X \mid M, \theta) \\ \nabla_\theta \log P(X \mid M, \theta) \end{bmatrix}. \tag{7}$$

The score-space space defined by this mapping is identical to the Fisher mapping with one extra dimension which consists of the log likelihood score itself. This mapping has the benefit that the performance of a classifier using these mappings will have a minimum test performance that equals the original generative model, *M*. The inclusion of the derivatives as "extra features" should give additional information for the classifier to use.

An alternative score-argument is the ratio of two generative models, M_1 and M_2,

$$f(\{p_k(X \mid M_k, \theta_k)\}) = \log \frac{P(X \mid M_1, \theta_1)}{P(X \mid M_2, \theta_2)} \tag{8}$$

where θ = [θ1 θ2]. The corresponding mapping using the first derivative score-operator is,

$$\Psi(X) = \nabla_\theta \log \frac{P(X \mid M_1, \theta_1)}{P(X \mid M_2, \theta_2)} \tag{9}$$

and using the first derivative with argument score-operator,

$$\Psi(X) = \begin{bmatrix} \log \dfrac{P(X \mid M_1, \theta_1)}{P(X \mid M_2, \theta_2)} \\ \nabla_\theta \log \dfrac{P(X \mid M_1, \theta_1)}{P(X \mid M_2, \theta_2)} \end{bmatrix}. \tag{10}$$

A likelihood ratio forces the classifier to model the class boundaries more accurately. The discrimination information encoded in the likelihood ratio score should also be in its derivatives.

D.3 Relevance Vector Machine

The classification problem still needs a better approach than SVM to generalize the model for sparse solutions. The use of a probabilistic Bayesian learning enables more sparse and accurate training [Tipping et al., 2001]. Some noted disadvantages of the support vector learning methodology are:

- SVM uses a large number of basis functions because the number of support vectors increases with the number of data sets [Burges et al., 1997].

- SVM does classify the class with a hyperplane, which is binary decision, but it would be better to predict the outputs based on the probabilistic methods. The posterior distribution, p(t|x) where t = target label of class, of the training data help to classify the unknown inputs [Tipping et al., 2001].

The RVM based on the probabilistic Bayesian approach overcomes the above limitations. The essence of an RVM is a fully probabilistic model with an automatic relevance determination prior over each model parameter. The sparseness in the RVM model is explicitly sought in a probabilistic model framework. The following section explains the framework of the RVM. A new approach to improve the RVM algorithm will be explained after framework of RVM section.

D.3.1 RVM Framework

The framework of RVM is originally due to Tipping [Tipping et al., 2001]. RVM and SVM share a similar framework as described in equation (11) below. The training output y is a linearly weighted sum with a basis function, $\Phi(x)$. The target function given each input data, $\{x_n, t_n\}_{n=1}$, is expressed by equation (12), and ϵ_n denotes the zero-mean Gaussian. In RVM, the estimating target function uses the Bayesian approach given the prior distribution over the weights for each hyperparameter. RVM requires the likelihood function over targets given weight parameter value, and the target needs to form a distribution to lessen the computational complexity. The target function is assumed to be logistic sigmoid function, and the distribution over target given weight forms the Bernoulli distribution like equation (13).

$$y(x; w) = \sum_{n=1} w_n \phi_n(x) = w^T \phi(x) \tag{11}$$

$$t_n = y(x_n; w) + \varepsilon_n \tag{12}$$

$$P(t \mid w) = \prod_{n=1} \sigma\{y(x_n; w)\}^{t_n} [1 - \sigma\{y(x_n; w)\}]^{1-t_n} \tag{13}$$

The solutions for equation (13) can be approximated by Laplace's method, which was first proposed by Mackay. The weight parameters are controlled by the individual hyperparameter to moderate the strength of the prior distribution. For fixed values of the hyperparamter of α, the weights indicate the mean value of the posterior distribution of the equation (14). Since $p(w|t, \alpha) \propto p(t|w)p(w|\alpha)$, the maximum value of weight parameter can be approximate by this relation.

$$\log\{p(t \mid w)\, p(w \mid \alpha)\} = \sum_{n=1} [t_n \log y_n + (1 - t_n) \log(1 - y_n)] - \frac{1}{2} w^T A w \text{ with } y = \sigma\{y(x_n; w)\} \tag{14}$$

By using Laplace's method, we obtain a quadratic approximation to the posterior distribution. The result of Laplace's method forms a Hessian matrix:

$$\nabla_w \nabla_w \log p(w \mid t, \alpha)\,|_{w_{mp}} = -(\Phi^T B \Phi + A) \tag{15}$$

where B is a diagonal matrix with variance of the target function, $B = \text{diag}(\beta_1, \beta_2, \ldots, \beta_N)$ with $\beta n = \sigma\{y(x_n)\}[1 - \sigma\{y(x_n)\}]$, $\phi_n(x) = K(x, x_n)$, and $A = \text{diag}(\alpha_1, \alpha_2, \ldots, \alpha_N)$. The Hessian matrix is then negated and inverted to find the covariance and mean of the Gaussian approximation using Cholesky decomposition method.

$$\Sigma = (\Phi^T B \Phi + A)^{-1} \tag{16}$$

$$w_{MP} = \Sigma \Phi^T B t \tag{17}$$

The covariance and weight parameter is approximated by the value of a hyperparamter α at each iteration. This is the way to train the model to find the covariance and the mean value of the input data. In the next section, we discuss how to improve the efficiency of the current RVM algorithm.

D.3.2 Towards Improving the Efficiency

Through the course of this project, we have investigated many ways to improve the efficiency of the RVM parameter estimation process. Most of these focused on active learning type approaches in which subsets of the data were identified and processed, and then the results of the estimates of these subsets were merged. Through this process, we were able to increase training set sizes two orders of magnitude to hundreds of thousands of vectors. However, this is still not adequate for speech recognition applications, where the ability to train on millions of vectors is required. Therefore, we began reexamining the computational efficiency of the training process.

The RVM training procedure attempts to reduce the unnecessary weight parameters in every iteration. With the large input data sets, the Cholesky decomposition step needs large amounts of memory and computation time to compute the inverse of the Hessian matrix. Tipping and Faul have defined a constructive approach where the model begins with only a single parameter specified [Tipping et al., 2003]. Parameters are then added to the system in a constructive fashion while still satisfying the original optimization function.

Li and Sung proposed the Sequential Bootstrapped SVM method [Li et al., 2005]. This method finds the convex hull in the given samples to reduce the size of the support vectors. They assumed the support

vectors are placed in the convex hull of each sample distributions on linearly separable classes. Since the RVM takes much computation to find the local optima with slow convergence, finding a convex hull from given sample may boost the convergence rate to find the local optima points.

One major problem that we faced during RVM training was that it involved a steepest descent optimization problem that required the use of Cholesky decomposition. This latter method requires that the original matrix be a positive definite matrix (i.e., the determinant of the matrix should be greater than zero, or equivalently all the eigenvalues be positive). However, numerical roundoff errors in it's entries can easily cause near-positive semi-definite matrices (i.e., matrices whose determinant though positive is very small) to be non-positive definite, and thus causing the training procedure to fail.

To resolve this issue, we implemented a modified Cholesky algorithm [Schnabel & Eskow, 1990]. This works by adding controlled amounts of noise to the matrix to make it positive definite. To minimize the effect of this noise on the optimization performance, it should be as small as possible. The modified Cholseky algorithm tries to achieve an optimal trade-off between producing a positive-definite matrix and perturbing the matrix entries to a minimal extent. It is based on the well-known Gershgorin circle theorem from numerical linear algebra that provides bounds on the location of eigenvalues as a function of the off-diagonal entries of the matrix [Horn & Johnson, 1990].

After this modification, our RVM training was quite robust and we were able to train models from any of our training data.

D.4 Experiments on Speech Recognition

RVMs have had significant success in several classification tasks. These tasks have, however, involved relatively small quantities of static data. Speech recognition, on the other hand, involves processing a very large amount of temporally evolving signals. In order to gain insight into the effectiveness of RVMs for speech recognition, we explored two tasks. We first experimented on the Deterding static vowel classification task which is a common benchmark used for new classifiers. Second, we applied the techniques described above to a complete small vocabulary recognition task. Comparison with SVM models are given below. For each task, the RVMs outperformed the SVM models both in terms of model sparsity and error rate.

In our first pilot experiment, we applied SVMs and RVMs to a publicly available vowel classification task, Deterding Vowels. This was a good data set to evaluate the efficacy of static classifiers on speech classification data since it has been used as a standard benchmark for several nonlinear classifiers for several years. In this evaluation, the speech data was collected at a 10 kHz sampling rate and low pass filtered at 4.7 kHz. The signal was then transformed to 10 log-area parameters, giving a 10 dimensional input space. A window duration of 50 msec was used for generating the features. The training set consisted of 528 frames from eight speakers and the test set consisted of 462 frames from a different set of seven speakers. The speech data consisted of 11 vowels uttered by each speaker in a h*d context. Though it appears to be a simple task, the small training set and significant confusion in the vowel data make it a very challenging task.

Table 1 shows the results for a range of nonlinear classification schemes on the Deterding vowel data. From the table, the SVM and RVM are both superior to nearly all other techniques. The RVM achieves performance rivaling the best performance reported on this data (30% error rate) while exceeding the error performance of SVMs and the best neural network classifier. Importantly, the RVM classifiers achieve superior performance to the SVM classifiers while utilizing nearly an order of magnitude fewer parameters. While we do not expect the superior error performance to be typical (on pure classification

tasks) we do expect the superior sparseness to be typical. This sparseness property is particularly important when attempting to build systems which are practical to train and test.

A hybrid recognition architecture was also developed that is a parallel of our SVM hybrid. Each phone-level classifier (either an SVM or RVM dichotomous classifier) is trained as a one-vs-all classifier. The classifiers are used to predict the probability of an acoustic segment. For the SVM hybrid, a sigmoid posterior fit is used to map the SVM distance to a probability. The RVM output is naturally probabilistic so no link function is needed.

The HMM system is used to generate alignments at the phone level. Each phone instance is treated as one segment. Since each segment could span a variable duration, we divide the segment into three regions in a set ratio and construct a composite vector from the mean vectors of the three regions. In our experiments empirical evidence showed that a 3-4-3 proportion generally gave optimal performance. The classifiers in our hybrid systems operate on composite vectors. For decoding, the segmentation information is obtained from a baseline HMM system —a cross-word triphone system with 8 Gaussian mixtures per state. Composite vectors are generated for each of the segments and posterior probabilities are hypothesized that are used to find the best word sequence using the Viterbi decoder. The HMM system also outputs a set of N-best hypotheses. The posterior probabilities for each hypothesis are determined and the most likely entry of the N-best list is produced.

The performance of RVMs on the static classification of vowel data gave us good reason to expect the performance on continuous speech would be appreciably better than that of the SVM system in terms of sparsity and on par with the SVM system in terms of accuracy. Our initial tests of this hypothesis have been on a telephone alphadigit task. Recent work on both alphabet and alphadigit systems has taken a focus on resolving the high rates of recognizer confusion for certain word sets. In particular, the E-set (B,C, D, E, G, P, T, V, Z, THREE) and A-set (A, J, K, H, EIGHT). The problems occur mainly because the acoustic differences between the letters of the sets are minimal. For instance, the letters B and D differ primarily in the first 10-20 ms during the consonant portion of the letter.

The OGI Alphadigit Corpus is a telephone database collected from approximately 3000 subjects. Each subject was a volunteer responding to a posting on the USEnet. The subjects were given a list of either 19 or 29 alphanumeric strings to speak. The strings in the lists were each six words long, and each list was "set up to balance phonetic context between all letter and digit pairs." There were 1102 separate prompting strings which gave a balanced coverage of vocabulary and contexts. The training, cross-validation and test sets consisted of 51544, 13926 and 3329 utterances respectively, each balanced for gender. The data sets have been chosen to make them speaker independent.

The hybrid SVM and RVM systems have been benchmarked on the OGI alphadigit corpus with a vocabulary of 36 words. A total of 29 phone models, one classifier per model, were used to cover the pronunciations. Each classifier was trained using the segmental features derived from 39-dimensional frame-level feature vectors comprised of 12 cepstral coefficients, energy, delta and acceleration coefficients. The full training set has as many as 30k training examples per classifier. However, the training routines employed for the RVM models are unable to utilize such a large set as mentioned earlier. The training set was, thus, reduced to 10,000 training examples per classifier (5,000 in-class and 5,000 out-of class).

The test set was an open-loop speaker independent set with 3329 sentences. The composite vectors are also normalized to the range -1 to 1 to assist in convergence of the SVM classifiers. Both the SVM and RVM hybrid systems use identical RBF kernels with the width parameter set to 0.5. The trade-off parameter for the SVM system was set to 50. The sigmoid posterior estimate for the SVM was constructed using a held-out set of nearly 14000 utterances. The results of the RVM and SVM systems are shown in Table 2. The important columns to notice in terms of performance are the error rate, average number of parameters and testing time. In all three, the RVM system outperforms the SVM system. It achieves a slightly better error rate of 14.8% compared to 15.5%. This error rate is obtained in over an order of magnitude fewer parameters. This naturally translates to well over an order of magnitude better runtime performance. However, the RVM does require significantly longer to train. Fortunately, that added training time is done off-line.

D.5 Experiments on Speaker Recognition

Speaker recognition is divided into two fundamental tasks: identification and verification. Identification involves determining who is speaking from a group of known speakers. It is often refered to as closed-set identification. In contrast, the verification is called as open-set verification because it distinguishes the claimed speaker from a group of unknown speakers [4]. We chose to evaluate SVMs and RVMs on speaker recognition because we had a baseline HMM system available. The performance of the SVM was compared to our HMM with GMM speaker recognition.

NIST 2001 speaker recognition evaluation data was used for all the experiments described in this section [NIST et al., 2003]. All utterances in the development data set were approximately 2 minutes in length. The development set contained 60 utterances for training and 78 utterances for testing. These utterances were taken from the Switchboard corpus. A standard 39-dimension MFCC feature vector was used.

The SVM classifier requires information about in-class and out-of class data for every speaker in the training set. Suppose a model ‘x’ has to be trained for utterance ‘x’, in which case the in-class data for training will contain all the 39 dimensional MFCC feature set for the utterance ‘x’, and the out-of-class data is obtained by randomly picking “n” feature vectors from all the remaining utterances in the training data set. The size of “n” was determined in such a way that the out-of-class data had twice the number of MFCC vectors when compared to the in-class data. This is an approximation and hence will not contain all the information required to represent the true out-of-class distribution, but this sort of approximation was necessary to make the SVM training computationally feasible. Hence, it has to be kept in mind that the performance of this system is based on classifiers that were exposed to only a small subset of data during training.

During testing, the test MFCC vectors are used as input to compute the distance using the functional form of the model. A distance is computed for every single test vector, and finally an average distance for the entire feature vector set is computed. The average distance is used for final decision making. An ideal decision threshold is zero for SVM classifiers, but for speaker verification tasks we can determine a

Approach	Word Error Rate	Avg # Parameters	Training Time	Testing Time
SVM: RBF Kernels	15.5%	994	3 hours	1.5 hours
RVM: RBF Kernels	14.8%	72	5 days	5 minutes

Table 1. Performance comparison of SVMs and RVMs on an alphadigit recognition data. The RVMs yield a large reduction in the parameter count while attaining superior performance.

threshold where the detection cost function is minimum (DCF) [NIST et al., 2003].

The first set of experiments was conducted to determine the optimum value of γ for the RBF kernel. It was observed that for γ values between 2.5 to 0.02 there was very little variation in the distance scores for the test utterances. Performance was stable between 0.03 and 0.01 as shown in the DET [Martin et al., 1997] curves of Figure 1. The minimum DCF points were obtained for each of these curves and it was observed that for γ =0.019 we obtained the lowest minimum DCF. The minimum DCF for various values of γ are shown in Table 1. The Equal Error Rate was 16% with a γ of 0.019 and the penalty parameter set to 50. It can be observed from the DET plot that there is very marginal change in performance for changes

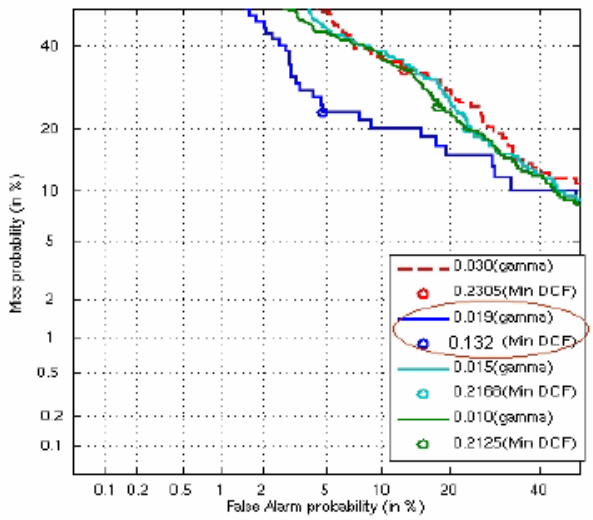

Figure 1. DET curves for various values of the RBF kernel parameter γ

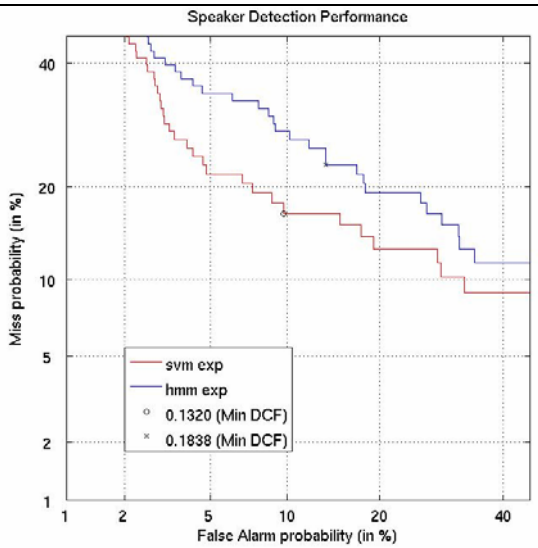

Figure 2. A comparison of HMM and SVM performance

in the γ values in the selected range. The most significant improvement in performance was observed only with a γ value of 0.019 and the effect of this improvement also reflected in an improvement in minimum DCF value as shown in Table 1.

We compared the results obtained on the SVM based speaker verification system with the baseline HMM system. The baseline system used 16-mixture Gaussians as the underlying classifier. An impostor model was trained on all the utterances in the development train set while the speaker models were built using the corresponding speaker utterance and constructing 16-mixture Gaussians. During testing, a likelihood ratio was computed between the speaker model and the impostor model. The likelihood ratio was defined as:

$$LR = \log P(x \mid sp\_\mathrm{mod}) - \log P(x \mid imp\_\mathrm{mod}) \qquad (18)$$

where LR is the likelihood ratio, "x" is the input test vector, "sp_mod" and "imp_mod" are the speaker and impostor models respectively. The equal error rate obtained on the HMM baseline system was close to 25% and the Min DCF was 0.1838. A comparative DET plot between SVM and baseline HMM system

Gamma(C=50)	Min DCF
0.010	0.2125
0.015	0.2168
0.019	0.1320
0.030	0.2305

Table 1. Minimum DCF as a function of γ

HMM	SVM
EER 25%	EER 16%
Min DCF 0.1838	Min DCF 0.1320

Table 2. Comparision of SVM based speaker verification system with the baseline HMM system

is shown in Figure 2 and their comparative performances are listed in Table 2.

D.6 Summary

This work is the first application of sparse Bayesian methods to speaker and continuous speech recognition. By using an automatic relevance determination mechanism, we are able to achieve state-of-the-art performance in extremely sparse models. Further, this is accomplished while maintaining a purely probabilistic framework. We also achieve performance better than the popular SVM kernel classifier while using an order of magnitude fewer parameters for both a static classification task and a continuous speech task. However, this runtime efficiency comes at a large up front cost during training. Thus, most of our work at this point is focused on more efficient training schemes so that we can move to larger vocabulary tasks. To this end, we have developed an iterative subset refinement approach which attempts to optimize the global criteria by locally optimizing the model on small subsets of the total training set. The subset models are incrementally used to generate a model of the full training set.

We are continuing our work on learning machines in speech recognition, and are now exploring new nonlinear statistical models under separate funding. This ITR project was our first opportunity to explore such risky and innovative methods.

E. References

L. R. Bahl, P. F. Brown, P. V. de Souza, and R. L. Mercer, "Maximum mutual information estimation of hidden Markov model parameters for speech recognition," *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, pp. 49-52, Tokyo, Japan, October 1986.

M. Beckman and J. Pierrehumbert, "Intonational structure in Japanese and English," Phonology Yearbook, 3, 255-309, 1986.

D. Bikel, On the Parameter Space of Lexicalized Statistical Parsing Models, Ph.D. thesis, University of Pennsylvania, 2004.

H.A. Bourlard and N. Morgan, *Connectionist Speech Recognition — A Hybrid Approach*, Kluwer Academic Publishers, Boston, USA, 1994.

C.J.C. Burges, B. Schölkopf, "Improving the accuracy and speed of support vector machines," Advances in Neural Information Processing Systems 9, pp 375-381, MIT Press, 1997.

C.J.C. Burges, A Tutorial on Support Vector Machines for Pattern Recognition, http:// svm.research.bell-labs.com/SVMdoc.html, AT&T Bell Labs, November 1999.

W. Byrne, M. Finke, S. Khudanpur, J. McDonough, H. Nock, M. Riley, M. Saraclar, C. Wooters and G. Zavaliagkos.
"Pronunciation Modelling Using a Hand-Labelled Corpus for Conversational Speech Recognition,"
IEEE International Conference on Acoustics, Speech and Signal Processing, Seattle, WA, 1998.

E. Charniak and M. Johnson, "Edit detection and parsing for transcribed speech," in Proc. NAACL 2001.

P. Clarkson and P. Moreno, "On the Use of Support Vector Machines for Phonetic Classification," *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, Phoenix, Arizona, USA, 1999.

R. Cole, "Alphadigit Corpus v1.0," *http://www.cse.ogi.edu/CSLU/corpora/alphadigit*, Center for Spoken Language Understanding, Oregon Graduate Institute, USA, 1998.

G.D. Cook and A.J. Robinson, "The 1997 ABBOT System for the Transcription of Broadcast News," *Proceedings of the DARPA Broadcast News Transcription and Understanding Workshop*, Lansdowne, VA, USA, 1998.

C. Cortes, V. Vapnik. Support Vector Networks, *Machine Learning*, vol. 20, pp. 293-297, 1995.

Y. Le Cun, L.D. Jackel, L. Bottou, C. Cortes, J. Denker, H. Drucker, I. Guyon, U.A. Muller, E. Sackinger, P. Simard, and V. Vapnik, "Learning algorithms for classification: A comparison on handwritten digit recognition," in *Neural Networks: The Statistical Mechanics Perspective*, J.H. Kwon and S. Cho, eds., pp. 261--276, World Scientific, 1995.

D. Deterding, M. Niranjan and A. J. Robinson, "Vowel Recognition (Deterding data)," Available at *http://www.ics.uci.edu/pub/machine-learning-databases/undocumented/connectionist-bench/vowel, 2000.*

A. C. Faul and M. E. Tipping, "Analysis of sparse Bayesian learning,". Neural Inform. Process. Syst., vol. 14, pp. 383–389, 2002.

A. Ganapathiraju, J. Hamaker and J. Picone, “Support Vector Machines for Speech Recognition,” *Proceedings of the International Conf. on Spoken Language Processing*, Sydney, Australia, Nov. 1998.

A. Ganapathiraju, J. Hamaker and J. Picone, “A Hybrid ASR System Using Support Vector Machines,” submitted to the *International Conf. of Spoken Language Processing*, Beijing, China, October, 2000.

A. Ganapathiraju, J. Hamaker and J. Picone, “Hybrid HMM/SVM Architectures for Speech Recognition,” *Proceedings of the Department of Defense Hub 5 Workshop*, College Park, Maryland, USA, May 2000.

A. Ganapathiraju, *Support Vector Machines for Speech Recognition*, Ph.D. Dissertation, Mississippi State University, Mississippi State, Mississippi, USA, 2002.

M. Gregory, M. Johnson, and E. Charniak, "Sentence-internal prosody does not help parsing the way punctuation does," in Proc. HLT-NAACL, 2004, pp. 81-88.

S. F. Gull, "Bayesian Inductive Inference and Maximum Entropy," *Maximum Entropy and Bayesian Methods in Science and Engineering, vol. 1, Foundations,* G. J. Erickson and C. R. Smith, eds., Kluwer Publishing, 1988.

J. Hamaker, J. Picone, and A. Ganapathiraju, "A Sparse Modeling Approach to Speech Recognition Based on Relevance Vector Machines," *Proceedings of the International Conference of Spoken Language Processing*, vol. 2, pp. 1001-1004, Denver, Colorado, USA, September 2002.

J. Hamaker, *Sparse Bayesian Methods for Continuous Speech Recognition*, Ph.D. Dissertation, Mississippi State University, Mississippi State, Mississippi, USA, 2003.

J. Hamaker and J. Picone, "Iterative Refinement of Relevance Vector Machines for Speech Recognition," submitted to *Eurospeech’03*, Geneva, Switzerland, September 2003.

D. Hindle, "Deterministic parsing of syntactic non-fluencies," in Proc. ACL, 1983, pp. 123-128.

T. Jaakkola, D. Haussler, “Exploiting Generative Models in Discriminative Classifiers,” Advances in Neural Information Processing Systems, vol. 11, MIT Press, 1999.

E. T. Jaynes, "Bayesian Methods: General Background," *Maximum Entropy and Bayesian Methods in Applied Statistics,* J. H. Justice, ed., pp. 1-25, Cambridge University Press, Cambridge, UK, 1986.

F. Jelinek, “Up From Trigrams! The Struggle for Improved Language Models,” *Proceedings of the 2nd European Conference on Speech Communication and Technology (Eurospeech)*, pp. 1037-1040, Genova, Italy, September 1991.

F. Jelinek, *Statistical Methods for Speech Recognition*, MIT Press, Cambridge, Massachusetts, USA, 1997.

T. Joachims, “Text Categorization with Support Vector Machines: Learning with Many Relevant Features,” *Technical Report 23, LS VIII, University of Dortmund*, Germany, 1997.

M. Johnson and E. Charniak, "A {TAG}-based noisy channel model of speech repairs," in Proc. ACL, 2004, pp. 33-39.

M. Johnson, E. Charniak, and M. Lease, “An improved model for recognizing disfluencies in conversational speech,” In Proc. Rich Text 2004 Fall Workshop (RT-04F).

J. Kahn, M. Ostendorf, and C. Chelba, "Parsing conversational speech using acoustic segmentation," in Proc. HLT-NAACL, comp. vol., 2004, pp. 125-128.

J. Kim, S. Schwarm and M. Ostendorf, "Detecting structural metadata with decision trees and transformation-based learning," in Proc. HLT-NAACL, pp. 137-144, May 2004.

Y. Liu, E. Shriberg, A. Stolcke and M. Harper, "Using machine learning to cope with imbalanced classes in natural speech: Evidence from sentence boundary and disfluency detection," in Proc. ICSLP, 2004.

X. Li, Y. Zhu, E. Sung, "Sequential Bootstrapped Support Vector Machines," Proceedings of International Joint Conference on Networks, July 2005.

Y. Liu, E. Shriberg, A. Stolcke, B. Peskin, J. Ang, D. Hillard, M. Ostendorf, M. Tomalin, P.C. Woodland and M. Harper, "Structural Metadata Research in the EARS Program," *Proc. ICASSP 2005, Volume V, pp. 957--960, March 2005 (Philadelphia, PA).*

P. Loizou and A. Spanias, "High-Performance Alphabet Recognition," *IEEE Transactions on Speech and Audio Processing*, pp. 430-445, 1996.

D. J. C. MacKay, *Bayesian Methods for Adaptive Models*, Ph. D. thesis, California Institute of Technology, Pasadena, California, USA, 1991.

D. J. C. MacKay, "Probable networks and plausible predictions --- a review of practical Bayesian methods for supervised neural networks," *Network: Computation in Neural Systems*, 6, pp. 469-505, 1995.

O. L. Mangasarian, W. Nick Street and W. H. Wolberg: ''Breast cancer diagnosis and prognosis via linear programming", *Operations Research*, 43(4), pp. 570-577, July-August 1995.

A. Martin, G. Doddington, M.Ordowski, M. Przybocki, "The DET curve in assessment of detection task performance," In Proceedings of Euro Speech, vol. 4, pp. 1895-1898, 1997.

L. Mayfield, M. Gavalda, Y. Seo, B. Suhm, W. Ward, and A. Waibel, "Parsing real input in {JANUS}: a concept-based approach," in Proc. TMI 95, 1995.

"NIST 2003 Speaker Recognition Evaluation Plan," http://www.nist.gov/speech/tests/spk/2003/doc/2003-spkrecevalplan-v2.2.pdf.

E. Noth, A. Batliner, A. Kiebling, R. Kompe, and H. Niemann, "Verbmobil: The Use of Prosody in the Linguistic Components of a Speech Understanding System," IEEE Trans. on Speech and Audio Processing, 8(5):519-532, 2000.

M. Ostendorf and S. Roukos, "A Stochastic Segment Model for Phoneme-Based Continuous Speech Recognition," *IEEE Trans. on Acoustics Speech and Signal Processing*, vol. 37, pp. 1857- 1867, 1989.

M. Ostendorf, V. Digalakis, and O. Kimball, "From HMM's to Segment Models: A Unified View of Stochastic Modeling for Speech Recognition," *IEEE Transactions on Speech and Audio Processing*, vol. 4, no. 5, pp. 360-378, 1996.

M. Ostendorf, "Linking Speech Recognition and Language Processing Through Prosody," CC-AI, vol. 15, no. 3, pp. 279-303, 1998.

M. Ostendorf, I. Shafran, S. Shattuck-Hufnagel, B. Byrne and L. Carmichael, "A prosodically labeled database of spontaneous speech," Proc. of the ISCA Workshop on Prosody in Speech Recognition and Understanding, pp. 119-121, October 2001.

E. Osuna, R. Freund, and F. Girosi, "Support Vector Machines: Training and Applications," *MIT AI Memo 1602*, March 1997.

J. Pitrelli, M. Beckman, and J. Hirschberg, "Evaluation of prosodic transcription labeling reliability in the ToBI framework," In Proc. of the International Conference on Spoken Language Processing, 1, 123-126, 1994.

P. J. Price, M. Ostendorf, S. Shattuck-Hufnagel and C. Fong, "The Use of Prosody in Syntactic Disambiguation," Journal of the Acoustical Society of America, vol. 90, no. 6, December 1991, pp. 2956-2970.

L. R. Rabiner, "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257-285, February 1989.

S. Renals, *Speech and Neural Network Dynamics*, Ph. D. dissertation, University of Edinburgh, UK, 1990.

M.D. Richard and R.P. Lippmann, "Neural Network Classifiers Estimate Bayesian a Posteriori Probabilities", *Neural Computation*, vol. 3, no. 4, pp. 461-483, 1991.

J. Rissanen, "Modeling by Shortest Data Description," *Automatica,* 14, pp. 465-471, 1978.

A. J. Robinson, *Dynamic Error Propagation Networks*, Ph.D. dissertation, Cambridge University, UK, February 1989.

B. Schölkopf, C. Burges and A. Smola, *Advances in Kernel Methods: Support Vector Machines*, MIT Press, Cambridge, MA, USA, December 1998.

G. Schwartz, "Estimating the Dimension of a Model," *Annals of Statistics,* vol. 6, no. 2, pp. 461-464, 1978.

I. Shafran, M. Ostendorf, and R. Wright, "Prosody and phonetic variability: lessons learned from acoustic model clustering," Proc. of the ISCA Workshop on Prosody in Speech Recognition and Understanding, pp. 127-131, October 2001.

E. Shriberg et al., "Prosody-based automatic segmentation of speech into sentences and topics," Speech Communication, 32(1-2), pp. 127-154, 2000.

D. Talkin, "Pitch Tracking," in Speech Coding and Synthesis, ed. W. B. Kleijn and K. K. Paliwal, Elsevier Science B.V., 1995.

J. Tebelskis, S*peech Recognition using Neural Networks*, Ph. D. dissertation, Carnegie Mellon University, Pittsburg, USA, 1995.

M. Tipping, "The Relevance Vector Machine," in *Advances in Neural Information Processing Systems*, S. Solla, T. Leen and K.-R. Muller, eds., pp. 652-658, MIT Press, 2000.

M. Tipping, "Sparse Bayesian Learning and the Relevance Vector Machine," *Journal of Machine Learning*, vol. 1, pp. 211-244, June 2001.

M. E. Tipping and A. Faul, "Fast Marginal Likelihood Maximisation for Sparse Bayesian Models," submitted to *Artificial Intelligence and Statistics '03,* 2003.

V.N. Vapnik, *Statistical Learning Theory*, John Wiley, New York, NY, USA, 1998.

V. Wan, "Speaker Verification using Support Vector Machines," University of Sheffield, Dissertation for Ph. D, 2003.

P. Woodland and D. Povey, "Very Large Scale MMIE Training for Conversational Telephone Speech Recognition," *Proceedings of the 2000 Speech Transcription Workshop,* University of Maryland, MD, USA, May 2000.

F. Appendix

03/31/05 — 09/30/06: RESEARCH AND EDUCATIONAL ACTIVITIES

This report covers a one-year period for which the original grant was extended. Due to administrative issues related to the starting date of the extension, though this report technically covers the period from 03/31/2005 to 09/30/2006, most of the work was performed from 01/01/2006 to 09/30/2006. During this period, staffing included two MS-level graduate students (J. Suh and S. Lee) who performed the research, one undergraduate (E. Trammel) who supported research activities and dissemination through the web, and PI summer salary to supervise the project.

This ITR award was a multi-institution award Johns Hopkins, Brown University, University of Washington, and Mississippi State University. MS State was the PI. For the extension period, only MS State contributed. MS State's role in the overall project was acoustic modeling for speech recognition. Specifically, we explored the use of new machine learning techniques, such as Support Vector Machines (SVM) and Relevance Vector Machines (RVM), in the context of a system that integrated natural language parsing and speech recognition.

Their were two main goals of the work during this period: (1) continue our explorations into improving the efficiency of SVM and RVM training; (2) apply these to a task less demanding in terms of training to further validate the impact of these techniques. For (1), our main focus was investigation of a relatively new approach to applying SVMs known as score-space kernels. We also examined new ways to estimate the Hessian matrix for RVMs, which is a major computational bottleneck. For (2), we examined the impact of this technology on a well-known speaker recognition task to which a number of machine learning algorithms have been applied. Both efforts are in progress at the time this report was generated.

G. Machine Learning for Speech Recognition

Generative methods such as a Hidden Markov models (HMM) combined with Gaussian Mixture models (GMM) have been the dominant method for acoustic modeling of speech. Though the performance of speech recognition has been improved based on these generative methods, these methods are limited in their ability to discriminate. Hence, we have focused on the development of a discriminative models based on Support Vector Machines (SVM) and Relevance Vector Machines (RVM). A new method has been introduced to boost generalization of the acoustic model. We have applied a probabilistic Bayesian learning machine termed the RVM as the core statistical modeling unit [1]. These algorithms are bein used in applications involving both speech and speaker recognition.

SVM is a relatively mature classification technique originally developed by Vapnik and his colleagues [1]. It has a good generalization ability which is achieved by deriving an optimal hyperplane with maximum margin between two classes. In many applications, the theory of SVM has been shown to provide higher performance than traditional learning machines and has been introduced as a powerful tool for solving classification problems [2]. Due to these advantages, the SVM has been applied to many classification or recognition fields, such as text categorization, object recognition, speaker verification, and face detection in images [3]. The support vector paradigm is based upon structural risk minimization (SRM) in which the learning process is posed as one of optimizing some risk function. The optimal learning machine is the one whose free parameters are set such that the risk is minimized [1].

However, SVMs still have two problems. First, while sparse, the size of the SVM models (number of non-zero weights) tends to scale linearly with the quantity of training data. Second, the SVMs are binary classifiers. We require a probabilistic classification which reflects the amount of uncertainty in our predictions [1]. In speech recognition this is an important disadvantage since there is significant overlap in the feature space which can not be modeled by a yes/no decision boundary. Thus, we require a probabilistic classification which reflects the amount of uncertainty in our predictions. The essence of an

Relevance Vector Machine (RVM) is a fully probabilistic model with an automatic relevance determination prior over each model parameter [1].

G.1 Support Vector Machine

SVM is powerful tool for distinguishing each class with a nonlinear system basis function. The fundamental idea of an SVM is to project the input space vectors to a high-dimensional feature space using a nonlinear map, which is defined as kernel function [5]. The Structural Risk Minimization (SRM) principle enables one to implement the SVM, since the SRM defines the boundary for training model errors and confidence interval via the VC dimension. The hyperplane will separate the class depending on the binary or n-class cases while SVM reduces the empirical risk. The power of SVMs lies in their ability to transform data to a higher dimensional space and construct a linear binary classifier in the higher dimensional space [6]. A linear hyperplane in the higher dimensional space transforms to a complex nonlinear decision region in the input feature space.

For improving the efficiency and performance of SVM, the score-space kernel has been investigated for computation and performance. Wan [5] has proposed the score-space kernel method that simulates the human disriminiation process. In terms of speaker recognition system, humans distinguish identity using the intonation, accent, and frequently occuring words. Since the speech utterance for one speaker is highly correlated between segments of speech, taking the score of whole utterance as factor will improve the system. The following sections briefly explain the score-space approach.

G.1.1 Generative Kernel Function

The kernel function is derived from a generative probability model [7]. Even though discriminative methods are proved to be superior to the generative models for classification problems, generative methods are excellent for extracting information from input features. Kernel methods are suitable for using discriminative classification with a generative probability model. Suppose, the training set is composed of X_i and the corresponding binary targets are Y_i. The targets of new training examples are obtained from a weighted sum of the training targets. The estimated targets are consisted of estimating of weights and kernel functions. The weights represent the overall importance of the each training example X_i, and the kernel function compute the closeness of the pair of datasets. The estimated target process is represented by:

$$\hat{Y} = sign(\sum_i Y_i w_i K(X_i, X)) \tag{1}$$

The kernel function should be derived by a probabilistic method. The probabilistic method measures the difference between input sample X_i and test sample X. The training targets can be assumed to have a logistic regression distribution, and the targets are estimated given input data X and parameter vector θ.

$$P(Y \mid X, \theta) = \sigma(Y\theta^T X) \quad \text{where } \sigma(y) = (1 + e^{-y})^{-1} \tag{2}$$

By assigning a distribution for θ like a zero mean Gaussian with a full covariance matrix ∑, the posterior distribution for training targets can reduce the complexity of model. The maximum a posteriori (MAP) estimate for the parameters θ given a training set of examples is found by maximizing the following penalized log likelihood:

$$\sum_i \log P(Y_i \mid X_i, \theta) + \log P(\theta) = \sum_i \log \sigma(Y_i \theta^T X_i) - \frac{1}{2}\theta^T \Sigma^{-1} \theta + c \tag{3}$$

where the constant c does not depend on θ. The similarity between data X_i and X can be captured by taking the gradient space of the model. The gradient of the generative model with respect to a parameter describes how that parameter contributes to the process of generating a particular data set. The posterior distribution over training targets are finally estimated by

$$P(Y \mid X, \theta) = \sigma(Y \sum_i Y_i w_i (X_i^T \Sigma X)) \tag{4}$$

Comparing the equation (1) and (4), the kernel function can de replaced by $K(X_i, X) = X_i^T \Sigma X$. Through these processes the generative properties are involved in kernel functions, and SVM is able to exploit the generative and discriminative properties at the same time.

G.1.2 Score-Space Kernels

We investigated a more specific kernel function which enables us to classify variable length sequences of input vectors in a space of fixed dimension, called the score-space [5]. The score-space kernel uses any parametric generative model to classify whole sequences. The space to which sequences are mapped is called the score-space, so named because it is defined by and derived from the likelihood score, p(X|M,θ) of a generative model M. Given a set of k generative models the generic formulation of the mapping of a sequence, X={$\mathbf{x}_1$, …,$\mathbf{x}_{Nl}$ }, to the score-space is

$$\Psi_{\hat{F}}^{f}(X) = \Psi_{\hat{F}} f(\{p_k(X \mid M_k, \theta_k)\}). \tag{5}$$

This equation consists of score-argument $f(\{p_k(X \mid \theta_k)\})$, which is a function of scores of a set of generative model, and score-mapping operator $\Psi_{\hat{F}}$, which maps the scalar score-argument to the score-space. Any function may be used as a score-argument. We deal with two specific cases that lead to the likelihood score-space kernel and the likelihood ratio score-space kernel. By setting the score-argument to be the log likelihood of a single generative model, M, parameterized by θ, and choosing the first derivative score-operator, we obtain the mapping for the likelihood score space.

$$\Psi(X) = \nabla_\theta \log P(X \mid M, \theta) \tag{6}$$

Each component of the score-space, $\Psi(X)$, corresponds to the derivative of the log likelihood score with respect to one of the parameters of the model. This mapping is known as the Fisher mapping. The gradient of the log likelihood with respect to a parameter describes how that parameter contributes to the process of generating a particular speaker model. For the exponential family of distributions, these gradients form sufficient statistics for the models. This gradient space also naturally preserves all the structural assumptions that the model encodes about the generation process. When the gradients are small then likelihood has reached a local maximum and vice versa.

Using the first derivative with argument score-operator and the same score-argument the mapping becomes

$$\Psi(X) = \begin{bmatrix} \log P(X \mid M, \theta) \\ \nabla_\theta \log P(X \mid M, \theta) \end{bmatrix}. \tag{7}$$

The score-space space defined by this mapping is identical to the Fisher mapping with one extra dimension which consists of the log likelihood score itself. This mapping has the benefit that the performance of a classifier using these mappings will have a minimum test performance that equals the original generative model, *M*. The inclusion of the derivatives as "extra features" should give additional information for the classifier to use.

An alternative score-argument is the ratio of two generative models, M_1 and M_2,

$$f(\{p_k(X \mid M_k, \theta_k)\}) = \log \frac{P(X \mid M_1, \theta_1)}{P(X \mid M_2, \theta_2)} \tag{8}$$

where θ = [θ1 θ2]. The corresponding mapping using the first derivative score-operator is,

$$\Psi(X) = \nabla_\theta \log \frac{P(X \mid M_1, \theta_1)}{P(X \mid M_2, \theta_2)} \tag{9}$$

and using the first derivative with argument score-operator,

$$\Psi(X) = \begin{bmatrix} \log \dfrac{P(X \mid M_1, \theta_1)}{P(X \mid M_2, \theta_2)} \\ \nabla_\theta \log \dfrac{P(X \mid M_1, \theta_1)}{P(X \mid M_2, \theta_2)} \end{bmatrix}. \tag{10}$$

A likelihood ratio forces the classifier to model the class boundaries more accurately. The discrimination information encoded in the likelihood ratio score should also be in its derivatives.

G.2 Relevance Vector Machine

The classification problem still needs a better approach than SVM to generalize the model for sparse solutions. The use of a probabilistic Bayesian learning enables more sparse and accurate training [8]. Some noted disadvantages of the support vector learning methodology are:

- SVM uses a large number of basis functions because the number of support vectors increases with the number of data sets [9].
- SVM does classify the class with a hyperplane, which is binary decision, but it would be better to predict the outputs based on the probabilistic methods. The posterior distribution, p(t|x) where t = target label of class, of the training data help to classify the unknown inputs [8].

The RVM based on the probabilistic Bayesian approach overcomes the above limitations. The essence of an RVM is a fully probabilistic model with an automatic relevance determination prior over each model parameter. The sparseness in the RVM model is explicitly sought in a probabilistic model framework. The following section explains the framework of the RVM. A new approach to improve the RVM algorithm will be explained after framework of RVM section.

G.2.1 RVM Framework

The framework of RVM is originally due to Tipping [8]. RVM and SVM share a similar framework as described in equation (11) below. The training output y is a linearly weighted sum with a basis function,

$\Phi(x)$. The target function given each input data, $\{x_n, t_n\}_{n=1}$, is expressed by equation (12), and ϵ_n denotes the zero-mean Gaussian. In RVM, the estimating target function uses the Bayesian approach given the prior distribution over the weights for each hyperparameter. RVM requires the likelihood function over targets given weight parameter value, and the target needs to form a distribution to lessen the computational complexity. The target function is assumed to be logistic sigmoid function, and the distribution over target given weight forms the Bernoulli distribution like equation (13).

$$y(x;w) = \sum_{n=1} w_n \phi_n(x) = w^T \phi(x) \tag{11}$$

$$t_n = y(x_n;w) + \varepsilon_n \tag{12}$$

$$P(t \mid w) = \prod_{n=1} \sigma\{y(x_n;w)\}^{t_n} [1 - \sigma\{y(x_n;w)\}]^{1-t_n} \tag{13}$$

The solutions for equation (13) can be approximated by Laplace's method, which was first proposed by Mackay. The weight parameters are controlled by the individual hyperparameter to moderate the strength of the prior distribution. For fixed values of the hyperparamter of α, the weights indicate the mean value of the posterior distribution of the equation (14). Since $p(w|t, \alpha) \propto p(t|w)p(w|\alpha)$, the maximum value of weight parameter can be approximate by this relation.

$$\log\{p(t \mid w)\, p(w \mid \alpha)\} = \sum_{n=1} [t_n \log y_n + (1 - t_n)\log(1 - y_n)] - \frac{1}{2} w^T A w \text{ with } y = \sigma\{y(x_n;w)\} \tag{14}$$

By using Laplace's method, we obtain a quadratic approximation to the posterior distribution. The result of Laplace's method forms a Hessian matrix:

$$\nabla_w \nabla_w \log p(w \mid t, \alpha)\,|_{w_{mp}} = -(\Phi^T B \Phi + A) \tag{15}$$

where B is a diagonal matrix with variance of the target function, $B = \text{diag}(\beta_1, \beta_2, \ldots, \beta_N)$ with $\beta n = \sigma\{y(x_n)\}[1 - \sigma\{y(x_n)\}]$, $\phi_n(x) = K(x, x_n)$, and $A = \text{diag}(\alpha_1, \alpha_2, \ldots, \alpha_N)$. The Hessian matrix is then negated and inverted to find the covariance and mean of the Gaussian approximation using Cholesky decomposition method.

$$\Sigma = (\Phi^T B \Phi + A)^{-1} \tag{16}$$

$$w_{MP} = \Sigma \Phi^T B t \tag{17}$$

The covariance and weight parameter is approximated by the value of a hyperparamter α at each iteration. This is the way to train the model to find the covariance and the mean value of the input data. In the next section, we discuss how to improve the efficiency of the current RVM algorithm.

G.2.2 Towards Improving the Efficiency

Through the course of this project, we have investigated many ways to improve the efficiency of the RVM parameter estimation process. Most of these focused on active learning type approaches in which subsets of the data were identified and processed, and then the results of the estimates of these subsets were merged. Through this process, we were able to increase training set sizes two orders of magnitude to hundreds of thousands of vectors. However, this is still not adequate for speech recognition applications,

where the ability to train on millions of vectors is required. Therefore, we began reexamining the computational efficiency of the training process.

The RVM training procedure attempts to reduce the unnecessary weight parameters in every iteration. With the large input data sets, the Cholesky decomposition step needs large amounts of memory and computation time to compute the inverse of the Hessian matrix. Tipping and Faul have defined a constructive approach where the model begins with only a single parameter specified [10]. Parameters are then added to the system in a constructive fashion while still satisfying the original optimization function.

Li and Sung proposed the Sequential Bootstrapped SVM method [11]. This method finds the convex hull in the given samples to reduce the size of the support vectors. They assumed the support vectors are placed in the convex hull of each sample distributions on linearly separable classes. Since the RVM takes much computation to find the local optima with slow convergence, finding a convex hull from given sample may boost the convergence rate to find the local optima points.

Our work on incorporating these methods is in progress. In parallel with this work, we decided to explore a speaker recognition application which was less demanding on the size of the training data, and could be used to diagnose problems with the core algorithms.

H. Experiments on Speaker Recognition

Speaker recognition is divided into two fundamental tasks: identification and verification. Identification involves determining who is speaking from a group of known speakers. It is often refered to as closed-set identification. In contrast, the verification is called as open-set verification because it distinguishes the claimed speaker from a group of unknown speakers [4]. We chose to evaluate SVMs and RVMs on speaker recognition because we had a baseline HMM system available. The performance of the SVM was compared to our HMM with GMM speaker recognition.

H.1 SVM Baseline

NIST 2001 speaker recognition evaluation data was used for all the experiments described in this section [12]. All utterances in the development data set were approximately 2 minutes in length. The development set contained 60 utterances for training and 78 utterances for testing. These utterances were taken from the Switchboard corpus. A standard 39-dimension MFCC feature vector was used.

The SVM classifier requires information about in-class and out-of class data for every speaker in the training set. Suppose a model ‘x’ has to be trained for utterance ‘x’, in which case the in-class data for training will contain all the 39 dimensional MFCC feature set for the utterance ‘x’, and the out-of-class data is obtained by randomly picking “n” feature vectors from all the remaining utterances in the training data set. The size of “n” was determined in such a way that the out-of-class data had twice the number of MFCC vectors when compared to the in-class data. This is an approximation and hence will not contain all the information required to represent the true out-of-class distribution, but this sort of approximation was necessary to make the SVM training computationally feasible. Hence, it has to be kept in mind that the performance of this system is based on classifiers that were exposed to only a small subset of data during training.

During testing, the test MFCC vectors are used as input to compute the distance using the functional form of the model. A distance is computed for every single test vector, and finally an average distance for the entire feature vector set is computed. The average distance is used for final decision making. An ideal decision threshold is zero for SVM classifiers, but for speaker verification tasks we can determine a threshold where the detection cost function is minimum (DCF) [12].

The first set of experiments was conducted to determine the optimum value of γ for the RBF kernel. It was observed that for γ values between 2.5 to 0.02 there was very little variation in the distance scores for the test utterances. Performance was stable between 0.03 and 0.01 as shown in the DET [13] curves of Figure 1. The minimum DCF points were obtained for each of these curves and it was observed that for γ =0.019 we obtained the lowest minimum DCF. The minimum DCF for various values of γ are shown in Table 1. The Equal Error Rate was 16% with a γ of 0.019 and the penalty parameter set to 50. It can be observed from the DET plot that there is very marginal change in performance for changes in the γ values

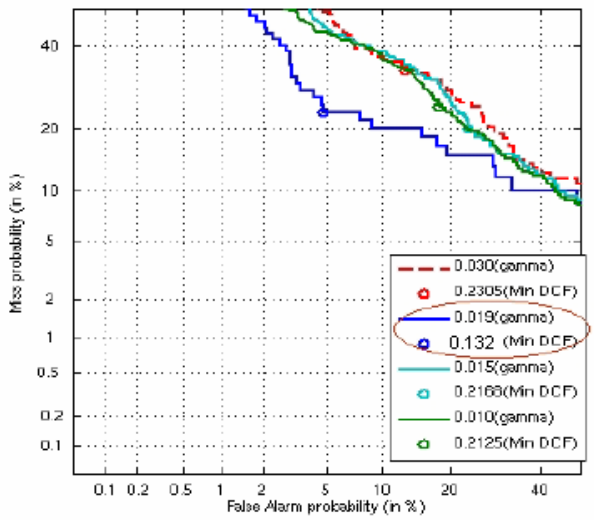

Figure 1. DET curves for various values of the RBF kernel parameter γ

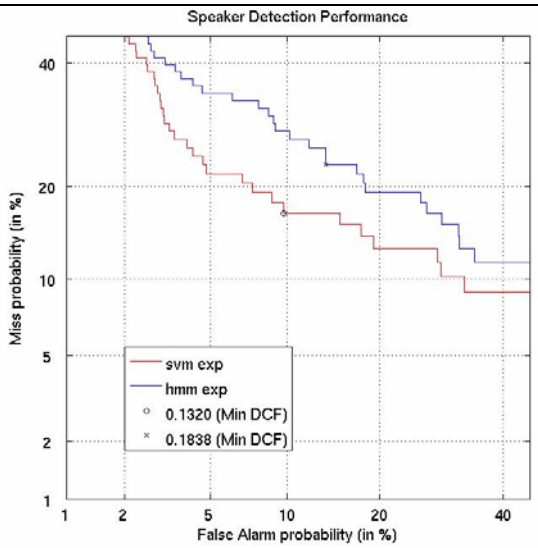

Figure 2. A comparison of HMM and SVM performance

in the selected range. The most significant improvement in performance was observed only with a γ value of 0.019 and the effect of this improvement also reflected in an improvement in minimum DCF value as shown in Table 1.

We compared the results obtained on the SVM based speaker verification system with the baseline HMM system. The baseline system used 16-mixture Gaussians as the underlying classifier. An impostor model was trained on all the utterances in the development train set while the speaker models were built using the corresponding speaker utterance and constructing 16-mixture Gaussians. During testing, a likelihood ratio was computed between the speaker model and the impostor model. The likelihood ratio was defined as:

$$LR = \log P(x \mid sp\_\mathrm{mod}) - \log P(x \mid imp\_\mathrm{mod}) \qquad (18)$$

where LR is the likelihood ratio, "x" is the input test vector, "sp_mod" and "imp_mod" are the speaker and impostor models respectively. The equal error rate obtained on the HMM baseline system was close to 25% and the Min DCF was 0.1838. A comparative DET plot between SVM and baseline HMM system is shown in Figure 2 and their comparative performances are listed in Table 2.

Gamma(C=50)	Min DCF
0.010	0.2125
0.015	0.2168
0.019	0.1320
0.030	0.2305

Table 1. Minimum DCF as a function of γ

HMM	SVM
EER 25%	EER 16%
Min DCF 0.1838	Min DCF 0.1320

Table 2. Comparision of SVM based speaker verification system with the baseline HMM system

I. Educational Activities

Our lab maintains a public domain speech recognition software environment and research infrastructure that has been a key mechanism for dissemination of information. We continued to upgrade our SVM and RVM capabilities as part of our general purpose toolkit available at: http://www.ece.msstate.edu/research/isip/projects/speech/software/. The SVM baseline system described in this report is available as part of this toolkit.

We also maintain a Java applet that encapsulates many pattern recognition principles at: http://www.ece.msstate.edu/research/isip/projects/speech/software/demonstrations/applets/util/pattern_recognition/current/index.html. This applet, which has been under development for several years, contains the most recent instantiation of our SVM and RVM training processes.

In addition to these resources, a number of lecture and presentation materials are available online at: http://www.ece.msstate.edu/research/isip/publications/seminars/. These document our intermediate progress and contain some tutorials on some of the underlying theory.

We conducted a individual study course in natural language processing in Fall'2005 that incorporated many of the concepts we were exposed to throughout this ITR project. Materials from this course are available at: http://www.ece.msstate.edu/research/isip/publications/courses/ece_7000_nlp/.

J. Conclusions

In this portion of the project, we have focused on improving the efficiency of learning machines such as SVMs and RVMs. SVM-based acoustic modeling requires less computation time for training and testing compared to an HMM in the speaker recognition problem. Unfortunately, training is more computationally demanding. The RVM requires more computation and memory for training, but improves classification performance. Work on improving the efficiency of an RVM continues.

The unexpected transition of our lab from the Center for Advanced Vechicular Systems to the Department of Electrical and Computer Engineering, which resulted in a loss of access to the university's supercomputing infrastructure, had a significant impact on our ability to run computationally-expensive RVM simulations. Our department has a very limited small cluster that is proving to be unstable. We also lost valuable time making this transition because we had to port our software and data infrastructure. This transition was completed over the summer, but the loss in computational infrastructure remains an issue.

K. Referencess

[1] J. Hamaker, J. Picone, "Advances in Speech Recognition Using Sparse Bayesian Methods," IEEE Transactions on Speech and Audio Processing, January 2003.

[2] V. N. Vapnik, "The Nature of Statistical Learning Theory," Springer, New York, 1995.

[3] S. Raghavan, G. Lazarou, J. Picone, "Speaker Verification Using Support Vector Machine," Proceedings of IEEE Southeast Conference, pp. 189-191, Memphis TN, March 2006.

[4] D.A. Reynolds, "Speaker Identification and Verification Using Gaussian Mixture Models", Speech Communication, vol. 17, pp. 91-108, 1995.

[5] V. Wan, "Speaker Verification using Support Vector Machines," University of Sheffield, Dissertation for Ph. D, 2003.

[6] A. Ganapathirju, “Support Vector Machines for Speech Recognition,” Ph. D. Dissertation, Depart ment of Electrical and Computer Engineering, Mississippi State University, January 2002.

[7] T. Jaakkola, D. Haussler, “Exploiting Generative Models in Discriminative Classifiers,” Advance s in Neural Information Processing Systems, vol. 11, MIT Press, 1999.

[8] M. E. Tipping, “Sparse Bayesian Learning and the Relevance Vector Machine,” Journal of Machi ne Learning Research 1, vol 211, 2001.

[9] C. J. C. Burges, B. Schölkopf, “Improving the accuracy and speed of support vector machines,” Advances in Neural Information Processing Systems 9, pp 375-381, MIT Press, 1997

[10] M. E. Tipping, A. Faul, “Fast Marginal Likelihood Maximization for Sparse Bayesian Models,” submitted to Artificial Intelligence and Statistics ’03, 2003

[11] X. Li, Y. Zhu, E. Sung, “Sequential Bootstrapped Support Vector Machines,” Proceedings of International Joint Conference on Networks, July 2005.

[12] “NIST 2003 Speaker Recognition Evaluation Plan,” *http://www.nist.gov/speech/tests/spk/2003/do c/2003-spkrecevalplan-v2.2.pdf*

[13] A. Martin, G. Doddington, M.Ordowski, M. Przybocki, “The DET curve in assessment of detecti on task performance,” In Proceedings of Euro Speech, vol. 4, pp. 1895-1898, 1997.

08/15/04 — 03/31/05: RESEARCH AND EDUCATIONAL ACTIVITIES

The overall goal of this ITR was to create a strong synergy between speech recognition (ASR) and natural language processing (NLP). At the time this project began, integration of ASR and NLP was in its infancy, particularly for conversational speech applications. Over the duration of this project, two significant things happened. First, through the parallel efforts of DoD-funded research, community-wide focus on converstaional speech was achieved. Progress was impressive as error rates on tasks such as Switchboard and Call Home English decreased from 50% to 10%. ASR technology was now producing transcripts that were useful to NLP systems, and could support information retrieval applications involving important quantities such as named entities.

Second, NLP research began to focus on the problem of parsing speech recognition output, which lacks punctuation and formatting that was previously considered crucial to high performance parsing. This latter issue was the main focus of this ITR, and to some extent served as a beacon for the community. We produced resources that were extremely valuable, such as the extensions to the Penn Treebank that were released in 2003 (reconciliation of the ISIP Switchboard segmentations and transcriptions with the Penn Treebank segmentations and transcriptions). We introduced the mainstream community to advanced statistical modeling techniques such as Support Vector Machines and enhanced these for NLP applications.

Further, in line with the primary goal of the ITR program, this project created close collaborations between groups who did not previously work together. The PIs collaborated on a number of new initiatives as offshoots of this project, including applications in parsing, information retrieval, and homeland security. A subset of the PIs participated in conversational speech evaluations and workshops (e.g., DARPA EARS). Hence, we can conclude that this project created new synergies and new research directions that will continue beyond the timeframe of this project.

In this final report, we briefly describe some of the significant findings of our research below.

A. Laboratory for Linguistic Information Processing, Brown University

Learning general functional dependencies, i.e. functions between arbitrary input and output spaces, is one of the main goals in supervised machine learning. Recent progress has to a large extent focused on designing flexible and powerful input representations, for instance by using kernel-based methods such as Support Vector Machines. We have addressed the complementary issue of problems involving complex outputs such as multiple dependent output variables and structured output spaces. In the context of this project we have mainly dealt with the problem of label sequence learning, a class of problems where dependencies between labels take the form of nearest neighbor dependencies along a chain or sequence of labels. The latter is a natural generalization of categorization or multiclass-classification that has many applications in the context of natural language processing and information extraction. Special cases include part-of-speech tagging, named entity recognition, and speech-accent prediction. More specifically, we have developed and empirically investigated several extensions of state-of-the-art categorization algorithms such as AdaBoost, Support Vector Machines, and Gaussian Process classification. We have designed and implemented several scalable learning algorithms that combine standard optimization techniques employed in the context of the above mentioned methods with dynamic programming techniques that account for the nearest neighbor dependencies. Experimental evaluations on a wide variety of tasks have shown the competitiveness of these methods compared to existing techniques like Hidden Markov Models and Conditional Random Fields.

A second line of research conducted in the context of the present ITR has dealt with ways to systematically exploit class hierarchies and taxonomies. The main question we have investigated is

whether or not a priori knowledge about the relationships between classes helps in improving classification accuracy, in particular in cases with many classes and few training examples. This is highly relevant for applications like word sense disambiguation and text categorization, where the number of classes can easily be in the tens of thousands. To that extend we have focused on a hierarchical version of the well-known perceptron learning algorithm as well as an extension of multiclass Support Vector Machines. We have shown that this approach can be effective in situations with sparse training data.

B. Center for Language and Speech Processing, Johns Hopkins University

The Structured Language Model (SLM) aims at making a prediction of the next word in a given word string by making a syntactical analysis of the preceding words. However, it faces the data sparseness problem because of the large dimensionality and diversity of the information available in the syntactic parses. A neural network model is better suited to tackle the data sparseness problem and its use has been shown to give significant improvements in perplexity and word error rate over the baseline SLM (Emami et al, 2003).

In this work we have investigated a new method of training the neural net based SLM. Our model makes use of a neural network for that component of the SLM that is responsible for predicting the next work given the previous words and their partial syntactic structure. We have investigated both a mismatched and a matched training scenario. In matched training, the neural network is trained on partial parses similar to those that are likely to be encountered during evaluation. On the other hand in the mismatched scenario, faster training time is achieved but at the cost of mismatch between training and evaluation and hence, possible degradation in performance.

The Structured Language Model works by assigning a probability P(W,T) to every sentence W and every possible binary parse T of W. The joint probability P(W,T) of a word sequence W and a complete parse T is broken into:

$$P(W,T) = \prod_{k=1}^{n+1} P(W_k \mid W_{k-1}T_{k-1}) \cdot P(t_k \mid W_{k-1}T_{k-1}, W_k) \cdot \prod_{i=1}^{N_k} P(p_i^k \mid W_{k-1}T_{k-1}, w_k, t_k, p_1^k \cdots p_{i-1}^k)$$

where $W_{k-1}T_{k-1}$ is the word-parse (k-1)-prefix, t_k is the tag assigned to w_k by the TAGGER, $N_k - 1$ is the number of operations the CONSTRUCTOR executes at sentence position k before passing control to the PREDICTOR, and p_i^k denotes the i-th CONSTRUCTOR operation carried out at position k in the word string.

Subsequently, the *language model* probability assignment for the word at position k+1 in the input sentence is made using:

$$\mathrm{P}_{\mathrm{SLM}}(w_{k+1} \mid W_k) = \sum_{T_k \in S_k} P(w_{k+1} \mid W_k T_k) \cdot \rho(W_k T_k)$$

$$\rho(W_k T_k) = P(W_k T_k) \Big/ \sum_{T_k \in S_k} P(W_k T_k)$$

which ensures a proper probability normalization over strings W^* where S_k is the set of all parses built and retained by the model at the current stage k.

Neural networks are very suitable for modeling conditional discrete distribution with large vocabularies. These models work by first assigning a continuous feature vector with every token in the vocabulary, and then using a standard multi-layered neural net to get the conditional distribution at the output, given the input feature vectors. Training is achieved by searching for parameters Θ of the neural network and the values of feature vectors that maximize the penalized log-likelihood of the training corpus:

$$L = \frac{1}{N}\sum_{t} \log p(y^t \mid x_1^t, x_2^t, \cdots, x_{n-1}^t; \Theta) - R(\Theta)$$

where $p(y^t \mid x_1^t, x_2^t, \cdots, x_{n-1}^t; \Theta)$ is the probability of word y^t (network output at time t), N is the training data size and R(Θ) is a regularization term, L-2 norm squared of the parameters in our case.

We have used a neural net to model the SCORER component of the SLM. By the SCORER we refer to the model $P(w_{k+1} \mid W_k T_k)$. The neural net SCORER's parameters can be obtained by training it on the events extracted from the gold standard (usually one best) parses obtained from an external source (humans or an automatic parser). However, there would be a mismatch during evaluation since the partial parses during that phase are not provided and have to be hypothesized by the SLM itself. We have called the SCORER trained in this manner the *mismatched* SCORER.

On the other hand, one can train the model on partial parses hypothesized by the baseline SLM, thus maximizing the proper log-likelihood function. We have called this procedure the *matched* training of the SCORER.

Experimental results have shown considerable improvement in both perplexity and WER when using a neural net based SLM, specially in the case of matched SCORER training. On the UPenn section of the WSJ corpus, perplexity reductions of 12% and 19% over the baseline SLM (with a perplexity of 132) have been observed when using the mismatched and matched neural net models respectively.

For the WER experiments, the neural net bases models were used to re-rank an N-best list output by a speech recognizer on the WSJ DARPA'93 HUB1 test set (with a 1-best WER of 13.7%). The mismatched and matched neural net models reduced the SLM baseline WER of 12.6% to 12.0% and 11.8% (for relative improvements of 4.8% and 6.3%) respectively.

In summary, neural network models showed to be capable of taking advantage of the richer probabilistic dependencies extracted through syntactic analysis. In our case the use of a neural net for the SCORER component of the Structured Language Model resulted in considerable improvements in both perplexity and Word Error Rate (WER) with the best results achieved when using a training procedure matched with the evaluation.

C. Signal, Speech, and Language Interpretation Lab, University of Washington

Prosody can be thought of as the "punctuation" in spoken language, particularly the indicators of phrasing and emphasis in speech. Most theories of prosody have a symbolic (phonological) representation for these events, but a relatively flat structure. In English, for example, two levels of prosodic phrases are usually distinguished: intermediate (minor) and intonational (major) phrases [Beckman & Pierrehumbert, 1986]. While there is evidence that both phrase-level emphasis (or, prominence) of words and prosodic phrases (perceived groupings of words) provide information for syntactic disambiguation [Price et al., 1991], the most important of these cues seems to be the prosodic phrases or the boundary events marking them. While prior work has looked at the use of prosody in automatic parsing of isolated sentences, a key component of our work involved sentence detection as well, since our goal is to handle continuous conversational speech. Hence, the focus of our work has been on automatically recognizing sentence boundaries and sentence-internal prosodic phrase structure and investigating methods for integrating that structure in parsing.

To support these efforts, we also worked on analysis of acoustic cues to prosodic structure. The most important (and best understood) acoustic correlates of prosody are continuous-valued, including fundamental frequency (F0), energy, and duration cues. Particularly at phrase boundaries, cues include significant falls or rises in fundamental frequency accompanied by word-final duration lengthening, energy drops, and optionally a silent pause. In addition, however, there is evidence of spectral cues to prosodic events, so some of our work explored these cues, which also have implications for improving speech recognition.

Our approach to integrating prosody in parsing is to use symbolic boundary events that have categorical perceptual differences, including prosodic break labels that build on linguistic notions of intonational phrases and hesitation phenomena but also higher level structure. These events are predicted from a combination of the continuous acoustic features, rather than using the acoustic features directly, because the intermediate representation simplifies training with high-level (sparse) structures. Just as most automatic speech recognition (ASR) systems use a small inventory of phones as an intermediate level between words and acoustic feature to have robust word models (especially for unseen words), the small set of word boundary events are also useful as a mechanism for generalizing the large number of continuous-valued acoustic features to different parse structures. This approach is currently somewhat controversial because of the high cost of hand labeling, and to some extent because of its association with a particular linguistic theory. However, the specific subset of labels used in this work are relatively theory neutral and language independent, and a key contribution of this work is the use of weakly supervised learning to reduce the cost of prosodic labeling.

An alternative approach, as in [Noth et al, 2000], is to assign categorical "prosodic" labels defined in terms of syntactic structure and presence of a pause, without reference to human perception, and automatically learn the association of other prosodic cues with these labels. While this approach has been very successful in parsing speech from human-computer dialogs, we expect that it will be problematic for conversational speech because of the longer utterance and potential confusion between fluent and disfluent pauses.

C.1 Data, Annotation and Development

The usefulness of speech databases for statistical analysis is substantially increased when the database is labeled for a range of linguistic phenomena, providing the opportunity to improve our understanding of the factors governing systematic suprasegmental and segmental variation in word forms. Prosodic labels and phonetic alignments are available for some read speech corpora, but only a few limited samples of spontaneous conversational speech have been prosodically labeled. To fill this gap, a subset of the

Switchboard corpus of spontaneous telephone-quality dialogs was labeled using a simplification of the ToBI system for transcribing pitch accents, boundary tones and prosodic constituent structure [Pitrelli et al., 1994]. The aim was to cover phenomena that would be most useful for automatic transcription and linguistic analysis of conversational speech. This effort was initiated under another research grant, but completed with partial support from this NSF ITR grant.

The corpus is based on about 4.7 hours of hand-labeled conversational speech (excluding silence and noise) from 63 conversations of the Switchboard corpus and 1 conversation from the CallHome corpus. The orthographic transcriptions are time-aligned to the waveform using segmentations available from Mississippi State University (http://www.cavs.msstate.edu/hse/ies/projects/switchboard/index.html) and a large vocabulary speech recognition system developed at the JHU Center for Language and Speech Processing for the 1997 Language Engineering Summer Workshop (www.clsp.jhu.edu/ws97) [Byrne et al., 1997]. All conversations were analyzed using a high quality pitch tracker [Talkin, 1995] to obtain F0 contours, then post-processed to eliminate errors due to crosstalk. The prosody transcription system included: i) breaks (0-4), which indicate the depth of the boundary after each word; ii) phrase prominence (none, *, *?); and iii) tones, which indicate syllable prominence and tonal boundary markers. It also provides a way to deal with the disfluencies that are common in spontaneous conversational speech (a p diacritic associated with the prosodic break), and to indicate labeler uncertainty about a particular transcription. The annotation does not include accent tone type, primarily to reduce transcription costs and because we hypothesized that the phrase tones would be relevant to dialog act representations which may be relevant for question-answering. For further information on the corpus and an initial distributional analysis, see [Ostendorf et al. 2001].

The prosodically labeled subset of Switchboard overlaps with the subset of that corpus annotated with Treebank parses, but there is a mismatch in the orthographic transcriptions because the Treebank parses were based on an earlier version of transcripts and the prosodic annotation was based on the higher quality corrections done by Prof. Picone's group (ISIP) at Mississippi State. In addition, we made use of the DARPA EARS metadata annotations that overlapped with the Treebank parses, which where again based on the higher quality transcriptions. To be able to use all of these resources, we used an alignment of words provided by the ISIP team, and mapped the Treebank parse information to the more recent word transcriptions, which could then be aligned with the EARS metadata annotations. Differences in transcriptions were handled by: dropping the parse information for deletions, transferring it as is for word substitutions, and treating it as "missing" information for insertions in the corrected transcripts. While most of the differences between the Treebank and corrected word transcriptions involved simple substitutions (or deletions) that had little or no impact on the parse (e.g. "a" vs. "the"), there were some cases where the transfer introduced noise into the collection of parses. The most frequent such cases were in disfluent regions, where transcribers tend to have more difficulties, including missed word fragments or repetitions ("I I" vs. "I I I"). An additional difference between the Treebank parses and the EARS metadata annotations is the marking of sentence boundaries. Since speakers frequently begin sentences with conjunctions, the metadata conventions often split up constituents marked as compound sentences in Treebank. Because the metadata labelers listened to the speech and the Treebank labelers did not, we chose to use the metadata constituents, which in most cases involved simply dropping a top-level (S) node, but in some cases involved adding a top-level node called "SUGROUP".

C.2 Automatic Labeling of Prosodic Structure

An important part of the effort was development of an automatic prosodic labeling system that would provide cues to improve parsing. In addition, the resulting system was inspected to analyze possible dependencies between prosodic and parse structures in conversational speech. In the experiments, we used decision tree classifiers with different combinations of acoustic, punctuation, parse, and disfluency cues. While more sophisticated techniques, such as HMMs and maximum entropy models, have been

used for related tasks of sentence boundary detection (see [Liu et al., 2005] for a brief survey), we chose decision trees because they are easy to inspect for learning about the prosody-syntax relationship and because this simplified the weakly supervised learning experiments, which were the focus of our efforts.

For the prosody/syntax analyses, we designed trees to predict prosodic labels from syntactic structure, as well as trees to predict prosodic structure from a combination of syntactic and acoustic cues. For purposes of providing information to a parser, we designed trees to predict prosodic constituents from acoustic cues and part-of-speech (POS) tags, but as an intermediate step in designing these trees we also used syntactic cues in designing trees as part of the weakly supervised training. More specifically, a small set of labeled data was used to train prosody models based on both text and acoustic cues, which were then used in combination to automatically label a large set of data that had not been hand-annotated with prosodic structure, and finally new (separate) acoustic-based prosody models were designed from this larger data set for use in parsing new data.

Experiments were conducted on the Switchboard corpus, using the prosodically annotated subset described above for initial training and evaluation (independent subsets for each). Then the full Switchboard training set was incorporated using various methods for weakly supervised learning, as described below. The prosodic constituent labels were merged into 3 classes: major intonational phrase boundary (4), hesitation boundary (1p, 2p), and all other fluent word boundaries. We grouped minor intonational phrase boundaries (3) with the default word boundary class, because preliminary experiments showed that they were almost never predicted by the decision trees (even with sampled training to account for the low frequency) and because they were most often confused with the default word class in 4-class prediction experiments. The simple 3-class system also has the advantage that it is relatively theory neutral and language independent in that essentially all languages have a notion of fluent and disfluent segmentation.

The acoustic cues included normalized F0, energy and duration cues based on those used in [Kim et al., 2004] and similar to those used in other metadata detection studies [Shriberg et al., 2000]. Text-based cues -- including punctuation, parse structure and disfluency markers -- were taken from the annotations available from the Linguistic Data Consortium, aligned to the word transcriptions as described above. The parse features included right-to-left and left-to-right relative depth, part-of-speech, label of the left and right syntactic constituents, and parenthetical markers. In addition, disfluency interruption points and flags for filled pauses and sentence-initial conjunctions were used as features. Punctuation as inserted by a human transcriber (including incomplete sentences) and estimated speaker turn boundaries (defined simply as a word boundary with a silence of length greater than 4s) were also used.

The baseline system trained on hand-labeled data has error rates of 9.6% when all available cues are used (both syntax and prosody) and 16.7% when just acoustic and POS cues are used. This can be compared to an error rate of 30% when the default class is assigned to all word boundaries. We considered three different weakly supervised training techniques for adding data without prosodic labels (but with hand-labeled syntactic structure) into the training set: EM, co-training, and self-training. The co-training algorithm used classifiers designed on either acoustic or syntactic cues, and it differed slightly from the standard method in that we used an information-theoretic distance on the tree posteriors to determine when to omit samples with conflicting classifier decisions. The self-training algorithm used bagging with uniform class sampling to deal with data skew [Liu et al., 2004]. In all cases, only 1-2 iterations were needed. Both the co-training and EM approaches gave improved performance over the baseline, with the

EM algorithm giving the best results of 14.2% error for the acoustic-only trees, which corresponds to a 15% reduction in error rate over supervised training. The self-training strategy actually hurt performance.[1]

From analysis of the resulting trees, we find that punctuation and repair points are correlated with prosodic breaks and hesitations, as expected. Silence duration is the most useful individual acoustic feature, but alone it is not a reliable predictor of prosodic phrases, since there are many prosodic phrases that do not occur at silences and because silences are frequently associated with hesitations. Aside from syntactic structure, the most important text features for predicting prosodic constituents are punctuation, disfluency edit point markers, filler words (sentence-initial coordinating conjunctions, discourse markers, filled pauses), and turn boundaries. Some important syntactic features include depth of subtrees on the left and right sides of the boundary, previous and next syntactic constituent tag, length of closing phrase, and part-of-speech tags. These features were relevant when associated with the target word boundary, but frequently also with the next or previous word boundary. Surprisingly, the label of the joining constituent is not useful. This analysis provided input into the parse reranking work described in the next section.

Due to the success of the weakly supervised training on prosodic phrase boundary detection, we have recently started investigating use of the same technique for training models of prosodic prominence. Initial results show only a 4% reduction in error rate for the system based on acoustic cues, from 22% to 21% error. Despite the high error rate, however, the automatically annotated prominence appears to be useful in topic identifications in preliminary experiments associated with a separate NSF project (IIS-0121396).

C.3 Prosody and Parsing

Parsing speech can be useful for a number of tasks, including information extraction and question answering from audio transcripts. However, parsing conversational speech presents a different set of challenges than parsing text: sentence boundaries are not well-defined, punctuation is absent, and presence of disfluencies (edits and restarts) impact the structure of language. Most prior work on parsing conversational speech has focused on handling disfluencies [Hindle 1983; Mayfield 1995; Charniak & Johnson, 2001], but experiments relied on hand-marked sentence boundaries and made use of punctuation as in text-based parsers. While utterance-level segmentation may be reasonable to assume in current human-computer dialog systems, it is not realistically available in recognized conversational speech. Hence, our work looked at the problem of parsing text with disfluencies and without punctuation.

We have investigated three main issues in the use of prosody in parsing: the impact of automatic sentence segmentation, the usefulness of interruption points, and the usefulness of automatically detected sub-sentence prosodic constituent boundaries (described above). In all cases, we use a two-stage architecture where metadata (constituent boundaries) are first detected with a combination of prosodic and simple text features, and then these symbolic events (or their posterior probabilities) are used in parsing. Our approach focuses on categorical boundary events, which are predicted from a combination of acoustic features, rather than using the acoustic features directly. As argued earlier, the intermediate representation simplifies training with sparse structures. Key research issues include whether the metadata should be treated as "words" or as features on words, whether edits should be represented with an independent component, and how to represent uncertainty of the metadata classifiers. Our work has begun investigating all of these questions, but some remain unanswered and are being pursued in ongoing work.

---

[1] The results reported here are in some cases worse than those reported in an earlier progress report, because they are based on a larger data set. Due to a data processing bug, several files were omitted from earlier studies. In addition, because of the larger amount of data used and the richer feature sets, the trees are much larger than those described in prior reports.

The data used in this work is the Treebank portion of the Switchboard corpus of conversational telephone speech, which includes sentence-like unit boundaries (SUs) as well as the reparandum and interruption point of disfluencies. The data consists of 816 hand-transcribed conversation sides (566K words), of which we reserve 128 conversation sides (61K words) for evaluation testing according to the 1993 NIST evaluation choices. In all cases, training was based on hand-labeled SUs. Parses were evaluated using SU boundaries rather than the standard punctuation-based units that the Treebank is based on, so the gold standard parses and parse evaluation metric were modified to incorporate the SUs.

The most exhaustive series of experiments looked at the impact of automatic segmentation on parsing. For this particular effort, we chose to work with the complete word sequence, i.e. including all of the words within edit regions, to allow experimentation with multiple parsers. In initial work [Kahn et al., 2004], we used the structured language model (SLM) as a parser with a simple pause-based segmentation and automatically detected SUs (69% vs. 35% slot error rate, respectively), showing a significant improvement in parsing performance when using the automatic SUs. We then confirmed the findings with results on the same segmentation alternatives but using two state-of-the-art statistical parsers trained on conversational speech: the Bikel [Bikel, 2004] [2] and Charniak [Charniak & Johnson, 2001] [3] parsers. Figure 1 shows the relative performance of each parser for the two different segmentations, comparing to the oracle performance for each using hand-annotated SUs. In order to have a single number for performance, we use the F-measure calculated from bracket precision and recall. (Trends with separate precision and recall measures and with bracket crossing statistics follow the same patterns.) For all three parsers, there is a significant gain in performance due to using the explicit SU detection algorithm, with more than half of the performance loss associated with the pause-based segmenter recovered when moving to the more sophisticated SU detection system. As SU detection improves, we would expect further performance gains. Also important is the observation that the impact of increased SU error is not lessened by using a better parser: the best oracle results were achieved with the Bikel parser, but it was much more sensitive to SU errors than the SLM.

In trying to understand the role of IPs in parsing, we ran several experiments, including some with and without human-annotated punctuation. Without punctuation, there was a small increase in parsing performance of the SLM using IPs When the SUs are automatically detected. We were not able to confirm these gains with other parsers; however, recent work in [Johnson, Charniak & Lease, 2004] shows a benefit to edit detection from using IPs which presumably would lead to improved parsing in their two-stage processing strategy [Johnson & Charniak, 2004]. Including punctuation and IPs in experiments with the SLM showed an

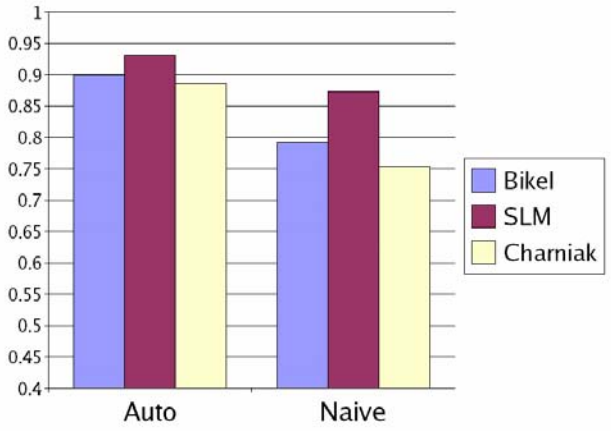

Figure 1. Relative performance of different parsers for the RT03 MDE SU detection system and a simple pause-based system, in each case comparing the F-measure to that obtained with oracle SUs.

[2] http://www.cis.upenn.edu/~dbikel/download.html (Version 0.9.9). For this work, we trained the Bikel parser on the Switchboard Treebank parses with the Collins settings.

[3] ftp://ftp.cs.brown.edu/pub/nlparser/ (August 2004)

interesting trade-off of bracket precision and recall using the two different cues, as illustrated in Figure 2. We saw the improved precision associated with using both punctuation and IPs as possible evidence that sub-sentence prosodic constituents might be useful.

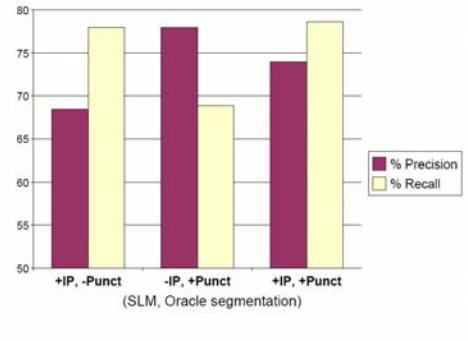

Figure 2. Precision-recall trade-offs associated with using punctuation, oracle IPs or both in addition to SUs in parsing.

In all of the above work, metadata is incorporated as "word" tokens, similar to the standard mechanism for parsers to incorporate punctuation. For sentence segmentation with a reasonably reliable segmenter, this may make sense, but certainly for sub-sentence prosodic constituents there is the potential for the gains associated with adding prosody to be offset by a loss from the extra words blocking part of the history that might be used in a statistical model of word dependence. We conjecture that this may in part explain the negative results obtained in [Gregory et al., 2004], since our analyses of the prosody prediction trees provides some evidence that sub-sentence prosodic constituents may be useful in parsing. (The direct use of acoustic features may also be problematic.) In addition, the use of metadata events as "words" requires a hard decision in the first stage of detection, and many results in speech processing suggest that soft decisions (e.g. using class posteriors) are more effective.

To address these problems, we developed an extension to the SLM that uses prosodic constituents as hidden conditioning variables, similar to headword conditioning in the SLM. However, since our subsequent work obtained much better baseline performance with other parsers, we decided to explore a parse reranking framework [Johnson et al., ms. in prep.] as an alternative method for incorporating automatically detected prosodic constituents. The approach uses a maximum entropy reranking model and introduces new features based on counts of syntactic constituent types weighted by the posterior probability of different prosodic events. Experiments with this new approach are in progress, now under other funding, and we anticipate having results in early 2005. This series of experiments will also look at the question of whether a separate stage of edit detection benefits parsing compared to simply incorporating the edit structure in the parser with the same status as other constituents.

C.4 Prosody and Acoustic Modeling

Most research on the use of prosody in automatic speech processing has focused on F0, energy and duration correlates to prosodic structure. However, there is evidence from long standing acoustic, articulatory and perceptual studies of speech suggesting that there are spectral correlates as well. For that reason, we conducted an analysis of our prosodically labeled conversational speech data using acoustic parameters and clustering techniques that are standard in speech recognition. We found that prosodic factors are associated with acoustic differences that can be learned in standard speech recognition systems. Both prosodic phrase structure and phrasal prominence seem to provide distinguishing cues, with some phones being affected much more than others (as one would expect from the linguistics literature). We hypothesized that we would find that constituent onsets were important at all levels (syllable, word and prosodic phrase). Instead, we found that onset is more important for syllables, but constituent-final position is more important at higher levels. Prosodic prominence had a smaller affect than phrase structure in terms of increasing likelihood of the training data, but seemed to result in more separable models when it did play a role.

Finally, we found evidence that segmental cues can help distinguish fluent from disfluent phrase boundaries, in that segments associated with these categories are frequently placed in different clusters. These differences can be leveraged in a "multiple pronunciation" acoustic model to aid in detecting fluent vs. disfluent prosodic boundaries, though additional prosodic cues are necessary to separate these from unmarked word boundaries. A limitation of this work was that it was based on hand-labeled data, and therefore did not take advantage of the full training data set needed for designing a state-of-the-art recognition system. However, with our recent developments in prosodic annotation, we will be able to assess the usefulness on a much larger corpus in the future.

D. Institute for Signal and Information Processing, Mississippi State University

Hidden Markov models (HMMs) with Gaussian emission densities are the prominent modeling technique in speech recognition. HMMs suffer from an inability to learn discriminative information and are prone to overfitting and overparameterization. Recent work in machine learning has focused on models, such as the support vector machine (SVM), that automatically control generalization and parameterization as part of the overall optimization process. SVMs, however, require ad hoc (and unreliable) methods to couple it to probabilistic speech recognition systems. We have applied a probabilistic Bayesian learning machine termed the relevance vector machine (RVM) as the core statistical modeling unit in a speech recognizer. The RVM is shown to provide superior performance compared to HMMs and SVMs in terms of both accuracy and sparsity on a continuous alphadigit task.

Computer speech recognition is typically posed as a statistical pattern recognition problem based on Bayesian methods in which the acoustic model and language model are treated as separate statistical models. The focus of our work has been the acoustic model, which maps sequences of features vectors to probabilities that these vectors were produced by a given linguistic unit, such as phone. In most state of the art recognition systems, a hidden Markov model (HMM) is used as the acoustic model. The popularity of the HMM representation is based on an HMMs ability to simultaneously model the temporal progression of speech (speech is usually seen as a "left-to-right" process) and the acoustic variability of the speech observations. The temporal variation is modeled via an underlying Markov process while the acoustic variability is modeled by an emission distribution at each state in the Markov chain.

The most commonly used emission distribution is the Gaussian mixture model (GMM). While combinations of HMMs and Gaussian mixture models have been extremely successful, there are two fundamental limitations of these approaches: (1) the parametric form of the underlying distribution is assumed to be Gaussian, (2) the maximum likelihood (ML) approach, which is typically used to estimate model parameters, does not explicitly improve the discriminative ability of the model. The ML approach maximizes the probability of the correct model while implicitly ignoring the probability of the incorrect model. Ideally, a training approach should force the model toward in-class training examples while simultaneously driving the model away from out-of-class training examples. Methods such as maximum mutual information and minimum classification error have been developed to incorporate discriminative training directly into the standard HMM/GMM framework. However, their success has been limited due primarily to their considerable computational costs.

The weaknesses of the HMM/GMM system have led researchers to explore other models, such as hybrid connectionist systems, which merge the power of artificial neural networks (ANNs) and HMMs. HMM/ANN hybrids have shown promise in terms of performance but have not yet found widespread use due to some serious problems. First, ANNs are prone to overfitting the training data if allowed to blindly converge. To avoid overfitting, a cross-validation set is often used to define a stopping point for training. This is wasteful of data and resources — a serious consideration in speech where the amount of labeled training data is limited. ANNs also typically converge much slower than HMMs. Most importantly, the

HMM/ANN hybrid systems have not shown the substantial improvements in recognition accuracy over HMM/GMM systems that would be necessary to force a paradigm shift.

The approaches developed in this work draw significantly from the HMM/ANN hybrid systems. However, we seek methods which are automatically immune to overfitting without the artificial imposition of a cross-validation set as well as methods which can automatically learn the appropriate model structure as part of the overall optimization process. One such model that has come to the forefront of pattern recognition research is the support vector machine (SVM). The support vector paradigm is based upon structural risk minimization (SRM) in which the learning process is posed as one of optimizing some risk function. The optimal learning machine is the one whose free parameters are set such that the risk is minimized.

Finding a minimum of the risk function is typically impossible due to the unknown distribution. Instead, it has been shown that a relationship exists between the actual risk, which is related to the empirical risk (i.e. the training set error which can be measured) and the Vapnik-Chervonenkis (VC) dimension. The VC dimension is a measure of the capacity of a learning machine to learn any training set and is typically closely related to the complexity of the learning machine's structure. With this result, we can guarantee both a small empirical risk (training error) and good generalization — an ideal situation for a learning machine.

In their most basic form SVMs use the SRM principle to impose an order on the optimization process by ranking candidate separating hyperplanes based on the margin they induce. For separable data, the optimal linear hyperplane is the one that maximizes the margin. The true power of the SVM, however, lies in how it deals with nonlinear class separating surfaces. Providing for a nonlinear decision region is accomplished using kernels. The optimization process yields a decision function where the sign of can be used to classify examples as either in-class or out-of-class. The decision function is formed from only those training vectors that lie on the margin or in overlap regions. In practice, the proportion of the training set that becomes support vectors is small, making the classifier sparse. Consequently, the training process, along with the training set, directly optimize the complexity of the learning machine. In contrast, ANN systems often make *a priori* assumptions about the form of the model.

SVMs have had great success on static classification tasks. However, it is only recently, that these techniques have been applied to continuous speech recognition. While the SVMs provide an excellent classification paradigm, they suffer from two serious drawbacks that hamper their effectiveness in speech recognition. First, while sparse, the size of the SVM models (number of non-zero weights) tends to scale linearly with the quantity of training data. For a large speaker independent corpus this effect is prohibitive. Second, the SVMs are binary classifiers. In speech recognition this is an important disadvantage since there is significant overlap in the feature space which can not be modeled by a yes/no decision boundary. Further, the combination of disparate knowledge sources (such as linguistic models, pronunciation models, acoustic models, etc.) requires a method for combining the scores produced by each model so that alternate hypotheses can be compared. Thus, we require a probabilistic classification which reflects the amount of uncertainty in our predictions.

We have investigated a Bayesian model termed the relevance vector machine (RVM) which is similar in form to the SVM but which addresses these two problems. The essence of an RVM is a fully probabilistic model with an automatic relevance determination prior over each model parameter. Thus, sparseness in the RVM model is explicitly sought in a probabilistic framework.

D.1 Sparse Bayesian Methods

Supervised learning in speech recognition implemented via a maximum likelihood approach is the dominant approach for finding values of the parameters in our model that best match the training data. Our expectation in data modeling is that given sufficient training data, the model would generalize to unseen test sets. Two levels of inference must be implemented to accomplish this. First, assuming that a particular model is true, we seek to infer the values for the parameters of the model that best fit the data at hand. This is exactly the training process used in the ANN hybrids. Second, we must decide which model is most appropriate given the data at hand, i.e. model comparison.

A simple approach to model comparison might dictate that we simply choose the model that fits the data best — the maximum likelihood solution. However, a more complex model can always fit the data better. Jaynes describes an extreme interpretation of this problem where we would always choose the so-called *Sure Thing* hypothesis, under which exactly the training set and only the training set is possible. Though it is the maximum likelihood solution, the Sure Thing hypothesis is intuitively displeasing and is counter to our desire for a solution which generalizes. We avoid choosing the Sure Thing hypothesis by expressing an *a priori* preference for simpler solutions using the principle of Occam's Razor. MacKay and others have formalized this preference mechanism through the use of Bayesian methods. These provide a natural and quantitative embodiment of Occam's razor. The first level of inference requires that we find the best-fit parameters. The second level of inference requires the comparison of competing hypotheses. If we assume that the competing hypotheses are *a priori* equiprobable then the best hypothesis is chosen by evaluating the evidence. The evidence is computed by marginalization across the model parameters.

The evidence provides a natural trade-off between the best-fit likelihood and the Occam factor. This concept is closely related to other methods such as the Minimum Description Length and the Bayesian Information Criteria where the model is directly penalized by the number of parameters used. A similar idea was also incorporated into SVM models, which penalize the models with too large a capacity (VC dimension). However, while the SVM models are forced to estimate the penalty via cross-validation schemes, Bayesian techniques automatically determine and apply the penalty in a fully probabilistic framework.

Assuming we have no prior knowledge that would cause us to favor a particular prior, we can find the optimal value for by evaluating the evidence. If we did have prior knowledge, we would simply repeat the inference over using the prior. At some level of the inference, we will arrive at a point where our prior knowledge is too weak to apply and then we evaluate the evidence. This extreme application of the Bayesian prior as a control parameter is known as the method of *automatic relevance determination* (ARD). It is so named because the prior over the input unit weights in a neural network can 'shut-off' those input dimensions which are irrelevant to the problem at hand. ARD is at the heart of the relevance vector machines that will be described next.

D.2 Relevance Vector Machines

An application of the evidence framework to kernel machines is the relevance vector machine (RVM). As with SVMs, RVMs use a weighted linear combination of basis functions. Due to the large number of parameters in this mode l— one per observation — we must guard against overfitting of the model to the training data. SVMs use a control parameter to implicitly balance the trade-off between training error and generalization. RVMs take a Bayesian approach and explicitly define an ARD prior distribution over the weights.

Each weight in the RVM model has an individual hyperparameter, , that is iteratively reestimated as part of the optimization process. As the hyperparameter grows larger, the prior on becomes infinitely peaked around zero, forcing to go to zero and, thus, contributing nothing to the summation. This process

automatically embodies the principle of Occam's Razor because it explicitly seeks the simplest model that satisfies the data constraints. In practice, the majority of the weights are pruned, resulting in an exceedingly sparse model with generalization abilities on par with SVMs. To complete the Bayesian specification of the model, we have to specify a prior probability. In practice we use a non-informative (flat) prior to indicate a lack of preference.

Approach	Error Rate	# Parameters
K-Nearest Neighbor	44%	
Gaussian Node Network	44%	
SVM: Polynomial Kernels	49%	
SVM: RBF Kernels	35%	83 SVs
Separable Mixture Models	30%	
RVM: RBF Kernels	30%	13 RVs

Table 1. Performance comparison of SVMs and RVMs to other nonlinear classifiers on static vowel classification data.

The parameter estimation process is an iterative reduction process. That is, initially each vector of the system is allocated one parameter. As the procedure continues, vectors are pruned from the model when they are found to be irrelevant with respect to the remaining parameters. Integral to this iterative reestimation process is the computation of the inverse Hessian matrix. This operation requires the inversion of an MxM Hessian matrix where M is initially set to the size of training set. For larger training sets (on the order of a few thousand), this computation is prohibitive both in time and memory.

D.3 Experiments

RVMs have had significant success in several classification tasks. These tasks have, however, involved relatively small quantities of static data. Speech recognition, on the other hand, involves processing a very large amount of temporally evolving signals. In order to gain insight into the effectiveness of RVMs for speech recognition, we explored two tasks. We first experimented on the Deterding static vowel classification task which is a common benchmark used for new classifiers. Second, we applied the techniques described above to a complete small vocabulary recognition task. Comparison with SVM models are given below. For each task, the RVMs outperformed the SVM models both in terms of model sparsity and error rate.

In our first pilot experiment, we applied SVMs and RVMs to a publicly available vowel classification task, Deterding Vowels. This was a good data set to evaluate the efficacy of static classifiers on speech classification data since it has been used as a standard benchmark for several nonlinear classifiers for several years. In this evaluation, the speech data was collected at a 10 kHz sampling rate and low pass filtered at 4.7 kHz. The signal was then transformed to 10 log-area parameters, giving a 10 dimensional input space. A window duration of 50 msec was used for generating the features. The training set consisted of 528 frames from eight speakers and the test set consisted of 462 frames from a different set of seven speakers. The speech data consisted of 11 vowels uttered by each speaker in a h*d context. Though it appears to be a simple task, the small training set and significant confusion in the vowel data make it a very challenging task.

Table 1 shows the results for a range of nonlinear classification schemes on the Deterding vowel data. From the table, the SVM and RVM are both superior to nearly all other techniques. The RVM achieves performance rivaling the best performance reported on this data (30% error rate) while exceeding the error performance of SVMs and the best neural network classifier. Importantly, the RVM classifiers achieve superior performance to the SVM classifiers while utilizing nearly an order of magnitude fewer parameters. While we do not expect the superior error performance to be typical (on pure classification tasks) we do expect the superior sparseness to be typical. This sparseness property is particularly important when attempting to build systems which are practical to train and test.

A hybrid recognition architecture was also developed that is a parallel of our SVM hybrid. Each phone-level classifier (either an SVM or RVM dichotomous classifier) is trained as a one-vs-all classifier. The classifiers are used to predict the probability of an acoustic segment. For the SVM hybrid, a sigmoid posterior fit is used to map the SVM distance to a probability. The RVM output is naturally probabilistic so no link function is needed.

The HMM system is used to generate alignments at the phone level. Each phone instance is treated as one segment. Since each segment could span a variable duration, we divide the segment into three regions in a set ratio and construct a composite vector from the mean vectors of the three regions. In our experiments empirical evidence showed that a 3-4-3 proportion generally gave optimal performance. The classifiers in our hybrid systems operate on composite vectors. For decoding, the segmentation information is obtained from a baseline HMM system —a cross-word triphone system with 8 Gaussian mixtures per state. Composite vectors are generated for each of the segments and posterior probabilities are hypothesized that are used to find the best word sequence using the Viterbi decoder. The HMM system also outputs a set of N-best hypotheses. The posterior probabilities for each hypothesis are determined and the most likely entry of the N-best list is produced.

The performance of RVMs on the static classification of vowel data gave us good reason to expect the performance on continuous speech would be appreciably better than that of the SVM system in terms of sparsity and on par with the SVM system in terms of accuracy. Our initial tests of this hypothesis have been on a telephone alphadigit task. Recent work on both alphabet and alphadigit systems has taken a focus on resolving the high rates of recognizer confusion for certain word sets. In particular, the E-set (B,C, D, E, G, P, T, V, Z, THREE) and A-set (A, J, K, H, EIGHT). The problems occur mainly because the acoustic differences between the letters of the sets are minimal. For instance, the letters B and D differ primarily in the first 10-20 ms during the consonant portion of the letter.

The OGI Alphadigit Corpus is a telephone database collected from approximately 3000 subjects. Each subject was a volunteer responding to a posting on the USEnet. The subjects were given a list of either 19 or 29 alphanumeric strings to speak. The strings in the lists were each six words long, and each list was “set up to balance phonetic context between all letter and digit pairs.” There were 1102 separate prompting strings which gave a balanced coverage of vocabulary and contexts. The training, cross-validation and test sets consisted of 51544, 13926 and 3329 utterances respectively, each balanced for gender. The data sets have been chosen to make them speaker independent.

The hybrid SVM and RVM systems have been benchmarked on the OGI alphadigit corpus with a vocabulary of 36 words. A total of 29 phone models, one classifier per model, were used to cover the pronunciations. Each classifier was trained using the segmental features derived from 39-dimensional frame-level feature vectors comprised of 12 cepstral coefficients, energy, delta and acceleration coefficients. The full training set has as many as 30k training examples per classifier. However, the training routines employed for the RVM models are unable to utilize such a large set as mentioned earlier. The training set was, thus, reduced to 10,000 training examples per classifier (5,000 in-class and 5,000 out-of class).

The test set was an open-loop speaker independent set with 3329 sentences. The composite vectors are also normalized to the range -1 to 1 to assist in convergence of the SVM classifiers. Both the SVM and RVM hybrid systems use identical RBF kernels with the width parameter set to 0.5. The trade-off parameter for the SVM system was set to 50. The sigmoid posterior estimate for the SVM was constructed using a held-out set of nearly 14000 utterances. The results of the RVM and SVM systems are shown in Table 2. The important columns to notice in terms of performance are the error rate, average number of parameters and testing time. In all three, the RVM system outperforms the SVM system. It achieves a slightly better error rate of 14.8% compared to 15.5%. This error rate is obtained in over an order of magnitude fewer parameters. This naturally translates to well over an order of magnitude better runtime performance. However, the RVM does require significantly longer to train. Fortunately, that added training time is done off-line.

D.4 Summary

This work is the first application of sparse Bayesian methods to continuous speech recognition. By using an automatic relevance determination mechanism, we are able to achieve state-of-the-art performance in extremely sparse models. Further, this is accomplished while maintaining a purely probabilistic framework. We also achieve performance better than the popular SVM kernel classifier while using an order of magnitude fewer parameters for both a static classification task and a continuous speech task. However, this runtime efficiency comes at a large up front cost during training. Thus, most of our work at this point is focused on more efficient training schemes so that we can move to larger vocabulary tasks. To this end, we have developed an iterative subset refinement approach which attempts to optimize the global criteria by locally optimizing the model on small subsets of the total training set. The subset models are incrementally used to generate a model of the full training set.

We are continuing our work on learning machines in speech recognition, and are now exploring new nonlinear statistical models under separate funding. This ITR project was our first opportunity to explore such risky and innovative methods.

Approach	Word Error Rate	Avg # Parameters	Training Time	Testing Time
SVM: RBF Kernels	15.5%	994	3 hours	1.5 hours
RVM: RBF Kernels	14.8%	72	5 days	5 minutes

Table 2. Performance comparison of SVMs and RVMs on Alphadigit recognition data. The RVMs yield a large reduction in the parameter count while attaining superior performance.

E. References

M. Beckman and J. Pierrehumbert, "Intonational structure in Japanese and English," Phonology Yearbook, 3, 255-309, 1986.

W. Byrne et al., "Pronunciation Modelling Using a Hand-Labelled Corpus for Conversational Speech Recognition," in Proc. ICASSP, 1998.

D. Bikel, On the Parameter Space of Lexicalized Statistical Parsing Models, Ph.D. thesis, University of Pennsylvania, 2004.

E. Charniak and M. Johnson, "Edit detection and parsing for transcribed speech," in Proc. NAACL 2001.

M. Gregory, M. Johnson, and E. Charniak, "Sentence-internal prosody does not help parsing the way punctuation does," in Proc. HLT-NAACL, 2004, pp. 81-88.

D. Hindle, "Deterministic parsing of syntactic non-fluencies," in Proc. ACL, 1983, pp. 123-128.

M. Johnson and E. Charniak, "A {TAG}-based noisy channel model of speech repairs," in Proc. ACL, 2004, pp. 33-39.

M. Johnson, E. Charniak and M. Lease, "An improved model for recognizing disfluencies n conversational speech," in Proc. NIST Rich Transcription Workshop, 2004, to appear.

J. Kahn, M. Ostendorf, and C. Chelba, "Parsing conversational speech using acoustic segmentation," in Proc. HLT-NAACL, comp. vol., 2004, pp. 125-128.

J. Kim, S. Schwarm and M. Ostendorf, "Detecting structural metadata with decision trees and transformation-based learning," in Proc. HLT-NAACL, pp. 137-144, May 2004.

Y. Liu, E. Shriberg, A. Stolcke and M. Harper, "Using machine learning to cope with imbalanced classes in natural speech: Evidence from sentence boundary and disfluency detection," in Proc. ICSLP, 2004.

Y. Liu et al., "Structural metadata research in the EARS program," in Proc. ICASSP, to appear, 2005.

L. Mayfield, M. Gavalda, Y. Seo, B. Suhm, W. Ward, and A. Waibel, "Parsing real input in {JANUS}: a concept-based approach," in Proc. TMI 95, 1995.

E. Noth, A. Batliner, A. Kiebling, R. Kompe, and H. Niemann, "Verbmobil: The Use of Prosody in the Linguistic Components of a Speech Understanding System," IEEE Trans. on Speech and Audio Processing, 8(5):519-532, 2000.

M. Ostendorf, "Linking Speech Recognition and Language Processing Through Prosody," CC-AI, vol. 15, no. 3, pp. 279-303, 1998.

M. Ostendorf, I. Shafran, S. Shattuck-Hufnagel, B. Byrne and L. Carmichael, "A prosodically labeled database of spontaneous speech," Proc. of the ISCA Workshop on Prosody in Speech Recognition and Understanding, pp. 119-121, October 2001.

J. Pitrelli, M. Beckman, and J. Hirschberg, "Evaluation of prosodic transcription labeling reliability in the ToBI framework," In Proc. of the International Conference on Spoken Language Processing, 1, 123-126, 1994.

P. J. Price, M. Ostendorf, S. Shattuck-Hufnagel and C. Fong, "The Use of Prosody in Syntactic Disambiguation," Journal of the Acoustical Society of America, vol. 90, no. 6, December 1991, pp. 2956-2970.

I. Shafran, M. Ostendorf, and R. Wright, "Prosody and phonetic variability: lessons learned from acoustic model clustering," Proc. of the ISCA Workshop on Prosody in Speech Recognition and Understanding, pp. 127-131, October 2001.

E. Shriberg et al., "Prosody-based automatic segmentation of speech into sentences and topics," Speech Communication, 32(1-2), pp. 127-154, 2000.

D. Talkin, "Pitch Tracking," in Speech Coding and Synthesis, ed. W. B. Kleijn and K. K. Paliwal, Elsevier Science B.V., 1995.

F. Jelinek, *Statistical Methods for Speech Recognition*, MIT Press, Cambridge, Massachusetts, USA, 1997.

F. Jelinek, "Up From Trigrams! The Struggle for Improved Language Models," *Proceedings of the 2nd European Conference on Speech Communication and Technology (Eurospeech)*, pp. 1037-1040, Genova, Italy, September 1991.

L. R. Rabiner, "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257-285, February 1989.

L. R. Bahl, P. F. Brown, P. V. de Souza, and R. L. Mercer, "Maximum mutual information estimation of hidden Markov model parameters for speech recognition," *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, pp. 49-52, Tokyo, Japan, October 1986.

P. Woodland and D. Povey, "Very Large Scale MMIE Training for Conversational Telephone Speech Recognition," *Proceedings of the 2000 Speech Transcription Workshop*, University of Maryland, MD, USA, May 2000.

S. Renals, *Speech and Neural Network Dynamics*, Ph. D. dissertation, University of Edinburgh, UK, 1990.

A. J. Robinson, *Dynamic Error Propagation Networks*, Ph.D. dissertation, Cambridge University, UK, February 1989.

J. Tebelskis, S*peech Recognition using Neural Networks*, Ph. D. dissertation, Carnegie Mellon University, Pittsburg, USA, 1995.

A. Ganapathiraju, *Support Vector Machines for Speech Recognition*, Ph.D. Dissertation, Mississippi State University, Mississippi State, Mississippi, USA, 2002.

M.D. Richard and R.P. Lippmann, "Neural Network Classifiers Estimate Bayesian a Posteriori Probabilities", *Neural Computation*, vol. 3, no. 4, pp. 461-483, 1991.

H.A. Bourlard and N. Morgan, *Connectionist Speech Recognition — A Hybrid Approach*, Kluwer Academic Publishers, Boston, USA, 1994.

G.D. Cook and A.J. Robinson, "The 1997 ABBOT System for the Transcription of Broadcast News," *Proceedings of the DARPA Broadcast News Transcription and Understanding Workshop*, Lansdowne, VA, USA, 1998.

V.N. Vapnik, *Statistical Learning Theory*, John Wiley, New York, NY, USA, 1998.

C. Cortes, V. Vapnik. Support Vector Networks, *Machine Learning*, vol. 20, pp. 293-297, 1995.

C.J.C. Burges, A Tutorial on Support Vector Machines for Pattern Recognition, http:// svm.research.bell-labs.com/SVMdoc.html, AT&T Bell Labs, November 1999.

B. Schölkopf, C. Burges and A. Smola, *Advances in Kernel Methods: Support Vector Machines*, MIT Press, Cambridge, MA, USA, December 1998.

O. L. Mangasarian, W. Nick Street and W. H. Wolberg: ‘‘Breast cancer diagnosis and prognosis via linear programming", *Operations Research*, 43(4), pp. 570-577, July-August 1995.

Y. Le Cun, L.D. Jackel, L. Bottou, C. Cortes, J. Denker, H. Drucker, I. Guyon, U.A. Muller, E. Sackinger, P. Simard, and V. Vapnik, "Learning algorithms for classification: A comparison on handwritten digit recognition," in *Neural Networks: The Statistical Mechanics Perspective*, J.H. Kwon and S. Cho, eds., pp. 261--276, World Scientific, 1995.

T. Joachims, "Text Categorization with Support Vector Machines: Learning with Many Relevant Features," *Technical Report 23, LS VIII, University of Dortmund*, Germany, 1997.

P. Clarkson and P. Moreno, "On the Use of Support Vector Machines for Phonetic Classification," *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, Phoenix, Arizona, USA, 1999.,

A. Ganapathiraju, J. Hamaker and J. Picone, "Support Vector Machines for Speech Recognition," *Proceedings of the International Conf. on Spoken Language Processing*, Sydney, Australia, Nov. 1998.

A. Ganapathiraju, J. Hamaker and J. Picone, "A Hybrid ASR System Using Support Vector Machines," submitted to the *International Conf. of Spoken Language Processing*, Beijing, China, October, 2000.

A. Ganapathiraju, J. Hamaker and J. Picone, "Hybrid HMM/SVM Architectures for Speech Recognition," *Proceedings of the Department of Defense Hub 5 Workshop*, College Park, Maryland, USA, May 2000.

M. Ostendorf, V. Digalakis, and O. Kimball, "From HMM's to Segment Models: A Unified View of Stochastic Modeling for Speech Recognition," *IEEE Transactions on Speech and Audio Processing*, vol. 4, no. 5, pp. 360-378, 1996.

M. Ostendorf and S. Roukos, "A Stochastic Segment Model for Phoneme-Based Continuous Speech Recognition," *IEEE Trans. on Acoustics Speech and Signal Processing*, vol. 37, pp. 1857- 1867, 1989.

M. Tipping, "Sparse Bayesian Learning and the Relevance Vector Machine," *Journal of Machine Learning*, vol. 1, pp. 211-244, June 2001.

M. Tipping, "The Relevance Vector Machine," in *Advances in Neural Information Processing Systems*, S. Solla, T. Leen and K.-R. Muller, eds., pp. 652-658, MIT Press, 2000.

D. J. C. MacKay, *Bayesian Methods for Adaptive Models*, Ph. D. thesis, California Institute of Technology, Pasadena, California, USA, 1991.

D. J. C. MacKay, "Probable networks and plausible predictions --- a review of practical Bayesian methods for supervised neural networks," *Network: Computation in Neural Systems*, 6, pp. 469-505, 1995.

E. T. Jaynes, "Bayesian Methods: General Background," *Maximum Entropy and Bayesian Methods in Applied Statistics*, J. H. Justice, ed., pp. 1-25, Cambridge University Press, Cambridge, UK, 1986.

S. F. Gull, "Bayesian Inductive Inference and Maximum Entropy," *Maximum Entropy and Bayesian Methods in Science and Engineering, vol. 1, Foundations*, G. J. Erickson and C. R. Smith, eds., Kluwer Publishing, 1988.

J. Rissanen, "Modeling by Shortest Data Description," *Automatica*, 14, pp. 465-471, 1978.

G. Schwartz, "Estimating the Dimension of a Model," *Annals of Statistics*, vol. 6, no. 2, pp. 461-464, 1978.

J. Hamaker, J. Picone, and A. Ganapathiraju, "A Sparse Modeling Approach to Speech Recognition Based on Relevance Vector Machines," *Proceedings of the International Conference of Spoken Language Processing*, vol. 2, pp. 1001-1004, Denver, Colorado, USA, September 2002.

E. Osuna, R. Freund, and F. Girosi, "Support Vector Machines: Training and Applications," *MIT AI Memo 1602*, March 1997.

A. Faul and M. Tipping, "Analysis of Sparse Bayesian Learning," *Proceedings of the Conference on Neural Information Processing Systems*, preprint, 2001.

J. Hamaker, *Sparse Bayesian Methods for Continuous Speech Recognition*, Ph.D. Dissertation, Mississippi State University, Mississippi State, Mississippi, USA, 2003.

M. E. Tipping and A. Faul, "Fast Marginal Likelihood Maximisation for Sparse Bayesian Models," submitted to *Artificial Intelligence and Statistics '03*, 2003.

D. Deterding, M. Niranjan and A. J. Robinson, "Vowel Recognition (Deterding data)," Available at *http://www.ics.uci.edu/pub/machine-learning-databases/undocumented/connectionist-bench/vowel*, *2000*.

P. Loizou and A. Spanias, "High-Performance Alphabet Recognition," *IEEE Transactions on Speech and Audio Processing*, pp. 430-445, 1996.

R. Cole, "Alphadigit Corpus v1.0," *http://www.cse.ogi.edu/CSLU/corpora/alphadigit*, Center for Spoken Language Understanding, Oregon Graduate Institute, USA, 1998.

J. Hamaker and J. Picone, "Iterative Refinement of Relevance Vector Machines for Speech Recognition," submitted to *Eurospeech'03*, Geneva, Switzerland, September 2003.

08/15/03 — 08/14/04: RESEARCH AND EDUCATIONAL ACTIVITIES

The overall goal of this ITR was to create a strong synergy between speech recognition (ASR) and natural language processing (NLP). In line with the primary goal of the ITR program, this project created close collaborations between groups who did not previously work together. The PIs collaborated on a number of new initiatives as offshoots of this project, including applications in parsing, information retrieval, and homeland security. In this annual report, we briefly describe some of the significant findings of our research below.

F. Laboratory for Linguistic Information Processing, Brown University

Learning general functional dependencies, i.e. functions between arbitrary input and output spaces, is one of the main goals in supervised machine learning. Recent progress has to a large extent focused on designing flexible and powerful input representations, for instance by using kernel-based methods such as Support Vector Machines. We have addressed the complementary issue of problems involving complex outputs such as multiple dependent output variables and structured output spaces. In the context of this project we have mainly dealt with the problem of label sequence learning, a class of problems where dependencies between labels take the form of nearest neighbor dependencies along a chain or sequence of labels. The latter is a natural generalization of categorization or multiclass-classification that has many applications in the context of natural language processing and information extraction. Special cases include part-of-speech tagging, named entity recognition, and speech-accent prediction. More specifically, we have developed and empirically investigated several extensions of state-of-the-art categorization algorithms such as AdaBoost, Support Vector Machines, and Gaussian Process classification. We have designed and implemented several scalable learning algorithms that combine standard optimization techniques employed in the context of the above mentioned methods with dynamic programming techniques that account for the nearest neighbor dependencies. Experimental evaluations on a wide variety of tasks have shown the competitiveness of these methods compared to existing techniques like Hidden Markov Models and Conditional Random Fields.

G. Center for Language and Speech Processing, Johns Hopkins University

The Structured Language Model (SLM) aims at making a prediction of the next word in a given word string by making a syntactical analysis of the preceding words. However, it faces the data sparseness problem because of the large dimensionality and diversity of the information available in the syntactic parses. A neural network model is better suited to tackle the data sparseness problem and its use has been shown to give significant improvements in perplexity and word error rate over the baseline SLM (Emami et al, 2003).

In this work we have investigated a new method of training the neural net based SLM. Our model makes use of a neural network for that component of the SLM that is responsible for predicting the next work given the previous words and their partial syntactic structure. We have investigated both a mismatched and a matched training scenario. In matched training, the neural network is trained on partial parses similar to those that are likely to be encountered during evaluation. On the other hand in the mismatched scenario, faster training time is achieved but at the cost of mismatch between training and evaluation and hence, possible degradation in performance.

The Structured Language Model works by assigning a probability P(W,T) to every sentence W and every possible binary parse T of W. The joint probability P(W,T) of a word sequence W and a complete parse T is broken into:

$$P(W,T) = \prod_{k=1}^{n+1} P(W_k \mid W_{k-1}T_{k-1}) \cdot P(t_k \mid W_{k-1}T_{k-1}, W_k) \cdot \prod_{i=1}^{N_k} P(p_i^k \mid W_{k-1}T_{k-1}, w_k, t_k, p_1^k \cdots p_{i-1}^k)$$

where $W_{k-1}T_{k-1}$ is the word-parse (k-1)-prefix, t_k is the tag assigned to w_k by the TAGGER, $N_k - 1$ is the number of operations the CONSTRUCTOR executes at sentence position k before passing control to the PREDICTOR, and p_i^k denotes the i-th CONSTRUCTOR operation carried out at position k in the word string.

Subsequently, the *language model* probability assignment for the word at position k+1 in the input sentence is made using:

$$\mathrm{P}_{\mathrm{SLM}}(w_{k+1} \mid W_k) = \sum_{T_k \in S_k} P(w_{k+1} \mid W_k T_k) \cdot \rho(W_k T_k)$$

$$\rho(W_k T_k) = P(W_k T_k) \Big/ \sum_{T_k \in S_k} P(W_k T_k)$$

which ensures a proper probability normalization over strings W^* where S_k is the set of all parses built and retained by the model at the current stage k.

Neural networks are very suitable for modeling conditional discrete distribution with large vocabularies. These models work by first assigning a continuous feature vector with every token in the vocabulary, and then using a standard multi-layered neural net to get the conditional distribution at the output, given the input feature vectors. Training is achieved by searching for parameters Θ of the neural network and the values of feature vectors that maximize the penalized log-likelihood of the training corpus:

$$L = \frac{1}{N} \sum_t \log p(y^t \mid x_1^t, x_2^t, \cdots, x_{n-1}^t; \Theta) - R(\Theta)$$

where $p(y^t \mid x_1^t, x_2^t, \cdots, x_{n-1}^t; \Theta)$ is the probability of word y^t (network output at time t), N is the training data size and R(Θ) is a regularization term, L-2 norm squared of the parameters in our case.

We have used a neural net to model the SCORER component of the SLM. By the SCORER we refer to the model $P(w_{k+1} \mid W_k T_k)$. The neural net SCORER's parameters can be obtained by training it on the events extracted from the gold standard (usually one best) parses obtained from an external source (humans or an automatic parser). However, there would be a mismatch during evaluation since the partial parses during that phase are not provided and have to be hypothesized by the SLM itself. We have called the SCORER trained in this manner the *mismatched* SCORER.

On the other hand, one can train the model on partial parses hypothesized by the baseline SLM, thus maximizing the proper log-likelihood function. We have called this procedure the *matched* training of the SCORER.

H. Signal, Speech, and Language Interpretation Lab, University of Washington

Prosody can be thought of as the "punctuation" in spoken language, particularly the indicators of phrasing and emphasis in speech. Most theories of prosody have a symbolic (phonological) representation for these events, but a relatively flat structure. In English, for example, two levels of prosodic phrases are usually distinguished: intermediate (minor) and intonational (major) phrases [Beckman & Pierrehumbert, 1986]. While there is evidence that both phrase-level emphasis (or, prominence) of words and prosodic phrases (perceived groupings of words) provide information for syntactic disambiguation [Price et al., 1991], the most important of these cues seems to be the prosodic phrases or the boundary events marking them. While prior work has looked at the use of prosody in automatic parsing of isolated sentences, a key component of our work involved sentence detection as well, since our goal is to handle continuous conversational speech. Hence, the focus of our work has been on automatically recognizing sentence boundaries and sentence-internal prosodic phrase structure and investigating methods for integrating that structure in parsing.

To support these efforts, we also worked on analysis of acoustic cues to prosodic structure. The most important (and best understood) acoustic correlates of prosody are continuous-valued, including fundamental frequency (F0), energy, and duration cues. Particularly at phrase boundaries, cues include significant falls or rises in fundamental frequency accompanied by word-final duration lengthening, energy drops, and optionally a silent pause. In addition, however, there is evidence of spectral cues to prosodic events, so some of our work explored these cues, which also have implications for improving speech recognition.

Our approach to integrating prosody in parsing is to use symbolic boundary events that have categorical perceptual differences, including prosodic break labels that build on linguistic notions of intonational phrases and hesitation phenomena but also higher level structure. These events are predicted from a combination of the continuous acoustic features, rather than using the acoustic features directly, because the intermediate representation simplifies training with high-level (sparse) structures. Just as most automatic speech recognition (ASR) systems use a small inventory of phones as an intermediate level between words and acoustic feature to have robust word models (especially for unseen words), the small set of word boundary events are also useful as a mechanism for generalizing the large number of continuous-valued acoustic features to different parse structures. This approach is currently somewhat controversial because of the high cost of hand labeling, and to some extent because of its association with a particular linguistic theory. However, the specific subset of labels used in this work are relatively theory neutral and language independent, and a key contribution of this work is the use of weakly supervised learning to reduce the cost of prosodic labeling.

An alternative approach, as in [Noth et al, 2000], is to assign categorical "prosodic" labels defined in terms of syntactic structure and presence of a pause, without reference to human perception, and automatically learn the association of other prosodic cues with these labels. While this approach has been very successful in parsing speech from human-computer dialogs, we expect that it will be problematic for conversational speech because of the longer utterance and potential confusion between fluent and disfluent pauses.

H.1 Data, Annotation and Development

The usefulness of speech databases for statistical analysis is substantially increased when the database is labeled for a range of linguistic phenomena, providing the opportunity to improve our understanding of the factors governing systematic suprasegmental and segmental variation in word forms. Prosodic labels and phonetic alignments are available for some read speech corpora, but only a few limited samples of spontaneous conversational speech have been prosodically labeled. To fill this gap, a subset of the

Switchboard corpus of spontaneous telephone-quality dialogs was labeled using a simplification of the ToBI system for transcribing pitch accents, boundary tones and prosodic constituent structure [Pitrelli et al., 1994]. The aim was to cover phenomena that would be most useful for automatic transcription and linguistic analysis of conversational speech. This effort was initiated under another research grant, but completed with partial support from this NSF ITR grant.

The corpus is based on about 4.7 hours of hand-labeled conversational speech (excluding silence and noise) from 63 conversations of the Switchboard corpus and 1 conversation from the CallHome corpus. The orthographic transcriptions are time-aligned to the waveform using segmentations available from Mississippi State University (http://www.cavs.msstate.edu/hse/ies/projects/switchboard/index.html) and a large vocabulary speech recognition system developed at the JHU Center for Language and Speech Processing for the 1997 Language Engineering Summer Workshop (www.clsp.jhu.edu/ws97) [Byrne et al., 1997]. All conversations were analyzed using a high quality pitch tracker [Talkin, 1995] to obtain F0 contours, then post-processed to eliminate errors due to crosstalk. The prosody transcription system included: i) breaks (0-4), which indicate the depth of the boundary after each word; ii) phrase prominence (none, *, *?); and iii) tones, which indicate syllable prominence and tonal boundary markers. It also provides a way to deal with the disfluencies that are common in spontaneous conversational speech (a p diacritic associated with the prosodic break), and to indicate labeler uncertainty about a particular transcription. The annotation does not include accent tone type, primarily to reduce transcription costs and because we hypothesized that the phrase tones would be relevant to dialog act representations which may be relevant for question-answering. For further information on the corpus and an initial distributional analysis, see [Ostendorf et al. 2001].

The prosodically labeled subset of Switchboard overlaps with the subset of that corpus annotated with Treebank parses, but there is a mismatch in the orthographic transcriptions because the Treebank parses were based on an earlier version of transcripts and the prosodic annotation was based on the higher quality corrections done by Prof. Picone's group (ISIP) at Mississippi State. In addition, we made use of the DARPA EARS metadata annotations that overlapped with the Treebank parses, which where again based on the higher quality transcriptions. To be able to use all of these resources, we used an alignment of words provided by the ISIP team, and mapped the Treebank parse information to the more recent word transcriptions, which could then be aligned with the EARS metadata annotations. Differences in transcriptions were handled by: dropping the parse information for deletions, transferring it as is for word substitutions, and treating it as "missing" information for insertions in the corrected transcripts. While most of the differences between the Treebank and corrected word transcriptions involved simple substitutions (or deletions) that had little or no impact on the parse (e.g. "a" vs. "the"), there were some cases where the transfer introduced noise into the collection of parses. The most frequent such cases were in disfluent regions, where transcribers tend to have more difficulties, including missed word fragments or repetitions ("I I" vs. "I I I"). An additional difference between the Treebank parses and the EARS metadata annotations is the marking of sentence boundaries. Since speakers frequently begin sentences with conjunctions, the metadata conventions often split up constituents marked as compound sentences in Treebank. Because the metadata labelers listened to the speech and the Treebank labelers did not, we chose to use the metadata constituents, which in most cases involved simply dropping a top-level (S) node, but in some cases involved adding a top-level node called "SUGROUP".

H.2 Automatic Labeling of Prosodic Structure

An important part of the effort was development of an automatic prosodic labeling system that would provide cues to improve parsing. In addition, the resulting system was inspected to analyze possible dependencies between prosodic and parse structures in conversational speech. In the experiments, we used decision tree classifiers with different combinations of acoustic, punctuation, parse, and disfluency cues. While more sophisticated techniques, such as HMMs and maximum entropy models, have been

used for related tasks of sentence boundary detection (see [Liu et al., 2005] for a brief survey), we chose decision trees because they are easy to inspect for learning about the prosody-syntax relationship and because this simplified the weakly supervised learning experiments, which were the focus of our efforts.

For the prosody/syntax analyses, we designed trees to predict prosodic labels from syntactic structure, as well as trees to predict prosodic structure from a combination of syntactic and acoustic cues. For purposes of providing information to a parser, we designed trees to predict prosodic constituents from acoustic cues and part-of-speech (POS) tags, but as an intermediate step in designing these trees we also used syntactic cues in designing trees as part of the weakly supervised training. More specifically, a small set of labeled data was used to train prosody models based on both text and acoustic cues, which were then used in combination to automatically label a large set of data that had not been hand-annotated with prosodic structure, and finally new (separate) acoustic-based prosody models were designed from this larger data set for use in parsing new data.

Experiments were conducted on the Switchboard corpus, using the prosodically annotated subset described above for initial training and evaluation (independent subsets for each). Then the full Switchboard training set was incorporated using various methods for weakly supervised learning, as described below. The prosodic constituent labels were merged into 3 classes: major intonational phrase boundary (4), hesitation boundary (1p, 2p), and all other fluent word boundaries. We grouped minor intonational phrase boundaries (3) with the default word boundary class, because preliminary experiments showed that they were almost never predicted by the decision trees (even with sampled training to account for the low frequency) and because they were most often confused with the default word class in 4-class prediction experiments. The simple 3-class system also has the advantage that it is relatively theory neutral and language independent in that essentially all languages have a notion of fluent and disfluent segmentation.

The acoustic cues included normalized F0, energy and duration cues based on those used in [Kim et al., 2004] and similar to those used in other metadata detection studies [Shriberg et al., 2000]. Text-based cues -- including punctuation, parse structure and disfluency markers -- were taken from the annotations available from the Linguistic Data Consortium, aligned to the word transcriptions as described above. The parse features included right-to-left and left-to-right relative depth, part-of-speech, label of the left and right syntactic constituents, and parenthetical markers. In addition, disfluency interruption points and flags for filled pauses and sentence-initial conjunctions were used as features. Punctuation as inserted by a human transcriber (including incomplete sentences) and estimated speaker turn boundaries (defined simply as a word boundary with a silence of length greater than 4s) were also used.

The baseline system trained on hand-labeled data has error rates of 9.6% when all available cues are used (both syntax and prosody) and 16.7% when just acoustic and POS cues are used. This can be compared to an error rate of 30% when the default class is assigned to all word boundaries. We considered three different weakly supervised training techniques for adding data without prosodic labels (but with hand-labeled syntactic structure) into the training set: EM, co-training, and self-training. The co-training algorithm used classifiers designed on either acoustic or syntactic cues, and it differed slightly from the standard method in that we used an information-theoretic distance on the tree posteriors to determine when to omit samples with conflicting classifier decisions. The self-training algorithm used bagging with uniform class sampling to deal with data skew [Liu et al., 2004]. In all cases, only 1-2 iterations were needed. Both the co-training and EM approaches gave improved performance over the baseline, with the

EM algorithm giving the best results of 14.2% error for the acoustic-only trees, which corresponds to a 15% reduction in error rate over supervised training. The self-training strategy actually hurt performance.[4]

From analysis of the resulting trees, we find that punctuation and repair points are correlated with prosodic breaks and hesitations, as expected. Silence duration is the most useful individual acoustic feature, but alone it is not a reliable predictor of prosodic phrases, since there are many prosodic phrases that do not occur at silences and because silences are frequently associated with hesitations. Aside from syntactic structure, the most important text features for predicting prosodic constituents are punctuation, disfluency edit point markers, filler words (sentence-initial coordinating conjunctions, discourse markers, filled pauses), and turn boundaries. Some important syntactic features include depth of subtrees on the left and right sides of the boundary, previous and next syntactic constituent tag, length of closing phrase, and part-of-speech tags. These features were relevant when associated with the target word boundary, but frequently also with the next or previous word boundary. Surprisingly, the label of the joining constituent is not useful. This analysis provided input into the parse reranking work described in the next section.

Due to the success of the weakly supervised training on prosodic phrase boundary detection, we have recently started investigating use of the same technique for training models of prosodic prominence. Initial results show only a 4% reduction in error rate for the system based on acoustic cues, from 22% to 21% error. Despite the high error rate, however, the automatically annotated prominence appears to be useful in topic identifications in preliminary experiments associated with a separate NSF project (IIS-0121396).

H.3 Prosody and Parsing

Parsing speech can be useful for a number of tasks, including information extraction and question answering from audio transcripts. However, parsing conversational speech presents a different set of challenges than parsing text: sentence boundaries are not well-defined, punctuation is absent, and presence of disfluencies (edits and restarts) impact the structure of language. Most prior work on parsing conversational speech has focused on handling disfluencies [Hindle 1983; Mayfield 1995; Charniak & Johnson, 2001], but experiments relied on hand-marked sentence boundaries and made use of punctuation as in text-based parsers. While utterance-level segmentation may be reasonable to assume in current human-computer dialog systems, it is not realistically available in recognized conversational speech. Hence, our work looked at the problem of parsing text with disfluencies and without punctuation.

We have investigated three main issues in the use of prosody in parsing: the impact of automatic sentence segmentation, the usefulness of interruption points, and the usefulness of automatically detected sub-sentence prosodic constituent boundaries (described above). In all cases, we use a two-stage architecture where metadata (constituent boundaries) are first detected with a combination of prosodic and simple text features, and then these symbolic events (or their posterior probabilities) are used in parsing. Our approach focuses on categorical boundary events, which are predicted from a combination of acoustic features, rather than using the acoustic features directly. As argued earlier, the intermediate representation simplifies training with sparse structures. Key research issues include whether the metadata should be treated as "words" or as features on words, whether edits should be represented with an independent component, and how to represent uncertainty of the metadata classifiers. Our work has begun investigating all of these questions, but some remain unanswered and are being pursued in ongoing work.

---

[4] The results reported here are in some cases worse than those reported in an earlier progress report, because they are based on a larger data set. Due to a data processing bug, several files were omitted from earlier studies. In addition, because of the larger amount of data used and the richer feature sets, the trees are much larger than those described in prior reports.

The data used in this work is the Treebank portion of the Switchboard corpus of conversational telephone speech, which includes sentence-like unit boundaries (SUs) as well as the reparandum and interruption point of disfluencies. The data consists of 816 hand-transcribed conversation sides (566K words), of which we reserve 128 conversation sides (61K words) for evaluation testing according to the 1993 NIST evaluation choices. In all cases, training was based on hand-labeled SUs. Parses were evaluated using SU boundaries rather than the standard punctuation-based units that the Treebank is based on, so the gold standard parses and parse evaluation metric were modified to incorporate the SUs.

The most exhaustive series of experiments looked at the impact of automatic segmentation on parsing. For this particular effort, we chose to work with the complete word sequence, i.e. including all of the words within edit regions, to allow experimentation with multiple parsers. In initial work [Kahn et al., 2004], we used the structured language model (SLM) as a parser with a simple pause-based segmentation and automatically detected SUs (69% vs. 35% slot error rate, respectively), showing a significant improvement in parsing performance when using the automatic SUs. We then confirmed the findings with results on the same segmentation alternatives but using two state-of-the-art statistical parsers trained on conversational speech: the Bikel [Bikel, 2004] [5] and Charniak [Charniak & Johnson, 2001] [6] parsers. Figure 1 shows the relative performance of each parser for the two different segmentations, comparing to the oracle performance for each using hand-annotated SUs. In order to have a single number for performance, we use the F-measure calculated from bracket precision and recall. (Trends with separate precision and recall measures and with bracket crossing statistics follow the same patterns.) For all three parsers, there is a significant gain in performance due to using the explicit SU detection algorithm, with more than half of the performance loss associated with the pause-based segmenter recovered when moving to the more sophisticated SU detection system. As SU detection improves, we would expect further performance gains. Also important is the observation that the impact of increased SU error is not lessened by using a better parser: the best oracle results were achieved with the Bikel parser, but it was much more sensitive to SU errors than the SLM.

In trying to understand the role of IPs in parsing, we ran several experiments, including some with and without human-annotated punctuation. Without punctuation, there was a small increase in parsing performance of the SLM using IPs When the SUs are automatically detected. We were not able to confirm these gains with other parsers; however, recent work in [Johnson, Charniak & Lease, 2004] shows a benefit to edit detection from using IPs which presumably would lead to improved parsing in their two-stage processing strategy [Johnson & Charniak, 2004]. Including punctuation and IPs in experiments with the SLM showed an

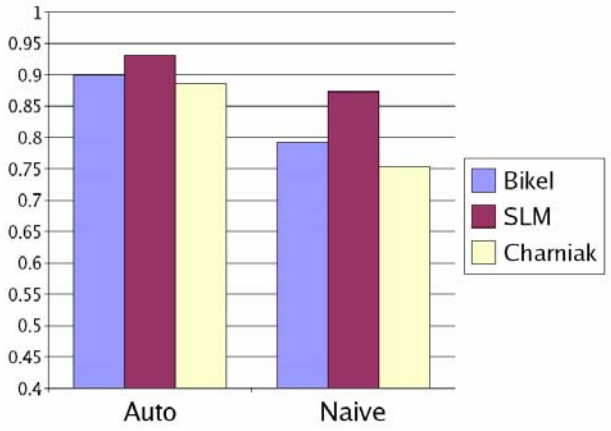

Figure 3. Relative performance of different parsers for the RT03 MDE SU detection system and a simple pause-based system, in each case comparing the F-measure to that obtained with oracle SUs.

[5] http://www.cis.upenn.edu/~dbikel/download.html (Version 0.9.9). For this work, we trained the Bikel parser on the Switchboard Treebank parses with the Collins settings.

[6] ftp://ftp.cs.brown.edu/pub/nlparser/ (August 2004)

interesting trade-off of bracket precision and recall using the two different cues, as illustrated in Figure 2. We saw the improved precision associated with using both punctuation and IPs as possible evidence that sub-sentence prosodic constituents might be useful.

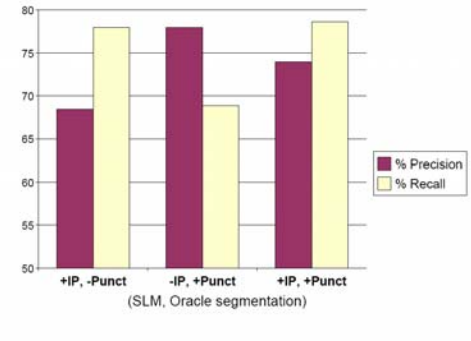

Figure 4. Precision-recall trade-offs associated with using punctuation, oracle IPs or both in addition to SUs in parsing.

In all of the above work, metadata is incorporated as "word" tokens, similar to the standard mechanism for parsers to incorporate punctuation. For sentence segmentation with a reasonably reliable segmenter, this may make sense, but certainly for sub-sentence prosodic constituents there is the potential for the gains associated with adding prosody to be offset by a loss from the extra words blocking part of the history that might be used in a statistical model of word dependence. We conjecture that this may in part explain the negative results obtained in [Gregory et al., 2004], since our analyses of the prosody prediction trees provides some evidence that sub-sentence prosodic constituents may be useful in parsing. (The direct use of acoustic features may also be problematic.) In addition, the use of metadata events as "words" requires a hard decision in the first stage of detection, and many results in speech processing suggest that soft decisions (e.g. using class posteriors) are more effective.

To address these problems, we developed an extension to the SLM that uses prosodic constituents as hidden conditioning variables, similar to headword conditioning in the SLM. However, since our subsequent work obtained much better baseline performance with other parsers, we decided to explore a parse reranking framework [Johnson et al., ms. in prep.] as an alternative method for incorporating automatically detected prosodic constituents. The approach uses a maximum entropy reranking model and introduces new features based on counts of syntactic constituent types weighted by the posterior probability of different prosodic events. Experiments with this new approach are in progress, now under other funding, and we anticipate having results in early 2005. This series of experiments will also look at the question of whether a separate stage of edit detection benefits parsing compared to simply incorporating the edit structure in the parser with the same status as other constituents.

I. Institute for Signal and Information Processing, Mississippi State University

Hidden Markov models (HMMs) with Gaussian emission densities are the prominent modeling technique in speech recognition. HMMs suffer from an inability to learn discriminative information and are prone to overfitting and overparameterization. Recent work in machine learning has focused on models, such as the support vector machine (SVM), that automatically control generalization and parameterization as part of the overall optimization process. SVMs, however, require ad hoc (and unreliable) methods to couple it to probabilistic speech recognition systems. We have applied a probabilistic Bayesian learning machine termed the relevance vector machine (RVM) as the core statistical modeling unit in a speech recognizer. The RVM is shown to provide superior performance compared to HMMs and SVMs in terms of both accuracy and sparsity on a continuous alphadigit task.

Computer speech recognition is typically posed as a statistical pattern recognition problem based on Bayesian methods in which the acoustic model and language model are treated as separate statistical models. The focus of our work has been the acoustic model, which maps sequences of features vectors to probabilities that these vectors were produced by a given linguistic unit, such as phone. In most state of

the art recognition systems, a hidden Markov model (HMM) is used as the acoustic model. The popularity of the HMM representation is based on an HMMs ability to simultaneously model the temporal progression of speech (speech is usually seen as a "left-to-right" process) and the acoustic variability of the speech observations. The temporal variation is modeled via an underlying Markov process while the acoustic variability is modeled by an emission distribution at each state in the Markov chain.

The most commonly used emission distribution is the Gaussian mixture model (GMM). While combinations of HMMs and Gaussian mixture models have been extremely successful, there are two fundamental limitations of these approaches: (1) the parametric form of the underlying distribution is assumed to be Gaussian, (2) the maximum likelihood (ML) approach, which is typically used to estimate model parameters, does not explicitly improve the discriminative ability of the model. The ML approach maximizes the probability of the correct model while implicitly ignoring the probability of the incorrect model. Ideally, a training approach should force the model toward in-class training examples while simultaneously driving the model away from out-of-class training examples. Methods such as maximum mutual information and minimum classification error have been developed to incorporate discriminative training directly into the standard HMM/GMM framework. However, their success has been limited due primarily to their considerable computational costs.

The weaknesses of the HMM/GMM system have led researchers to explore other models, such as hybrid connectionist systems, which merge the power of artificial neural networks (ANNs) and HMMs. HMM/ANN hybrids have shown promise in terms of performance but have not yet found widespread use due to some serious problems. First, ANNs are prone to overfitting the training data if allowed to blindly converge. To avoid overfitting, a cross-validation set is often used to define a stopping point for training. This is wasteful of data and resources — a serious consideration in speech where the amount of labeled training data is limited. ANNs also typically converge much slower than HMMs. Most importantly, the HMM/ANN hybrid systems have not shown the substantial improvements in recognition accuracy over HMM/GMM systems that would be necessary to force a paradigm shift.

The approaches developed in this work draw significantly from the HMM/ANN hybrid systems. However, we seek methods which are automatically immune to overfitting without the artificial imposition of a cross-validation set as well as methods which can automatically learn the appropriate model structure as part of the overall optimization process. One such model that has come to the forefront of pattern recognition research is the support vector machine (SVM). The support vector paradigm is based upon structural risk minimization (SRM) in which the learning process is posed as one of optimizing some risk function. The optimal learning machine is the one whose free parameters are set such that the risk is minimized.

Finding a minimum of the risk function is typically impossible due to the unknown distribution. Instead, it has been shown that a relationship exists between the actual risk, which is related to the empirical risk (i.e. the training set error which can be measured) and the Vapnik-Chervonenkis (VC) dimension. The VC dimension is a measure of the capacity of a learning machine to learn any training set and is typically closely related to the complexity of the learning machine's structure. With this result, we can guarantee both a small empirical risk (training error) and good generalization — an ideal situation for a learning machine.

In their most basic form SVMs use the SRM principle to impose an order on the optimization process by ranking candidate separating hyperplanes based on the margin they induce. For separable data, the optimal linear hyperplane is the one that maximizes the margin. The true power of the SVM, however, lies in how it deals with nonlinear class separating surfaces. Providing for a nonlinear decision region is accomplished using kernels. The optimization process yields a decision function where the sign of can be used to classify examples as either in-class or out-of-class. The decision function is formed from only

those training vectors that lie on the margin or in overlap regions. In practice, the proportion of the training set that becomes support vectors is small, making the classifier sparse. Consequently, the training process, along with the training set, directly optimize the complexity of the learning machine. In contrast, ANN systems often make *a priori* assumptions about the form of the model.

SVMs have had great success on static classification tasks. However, it is only recently, that these techniques have been applied to continuous speech recognition. While the SVMs provide an excellent classification paradigm, they suffer from two serious drawbacks that hamper their effectiveness in speech recognition. First, while sparse, the size of the SVM models (number of non-zero weights) tends to scale linearly with the quantity of training data. For a large speaker independent corpus this effect is prohibitive. Second, the SVMs are binary classifiers. In speech recognition this is an important disadvantage since there is significant overlap in the feature space which can not be modeled by a yes/no decision boundary. Further, the combination of disparate knowledge sources (such as linguistic models, pronunciation models, acoustic models, etc.) requires a method for combining the scores produced by each model so that alternate hypotheses can be compared. Thus, we require a probabilistic classification which reflects the amount of uncertainty in our predictions.

We have investigated a Bayesian model termed the relevance vector machine (RVM) which is similar in form to the SVM but which addresses these two problems. The essence of an RVM is a fully probabilistic model with an automatic relevance determination prior over each model parameter. Thus, sparseness in the RVM model is explicitly sought in a probabilistic framework.

I.1 Sparse Bayesian Methods

Supervised learning in speech recognition implemented via a maximum likelihood approach is the dominant approach for finding values of the parameters in our model that best match the training data. Our expectation in data modeling is that given sufficient training data, the model would generalize to unseen test sets. Two levels of inference must be implemented to accomplish this. First, assuming that a particular model is true, we seek to infer the values for the parameters of the model that best fit the data at hand. This is exactly the training process used in the ANN hybrids. Second, we must decide which model is most appropriate given the data at hand, i.e. model comparison.

A simple approach to model comparison might dictate that we simply choose the model that fits the data best — the maximum likelihood solution. However, a more complex model can always fit the data better. Jaynes describes an extreme interpretation of this problem where we would always choose the so-called *Sure Thing* hypothesis, under which exactly the training set and only the training set is possible. Though it is the maximum likelihood solution, the Sure Thing hypothesis is intuitively displeasing and is counter to our desire for a solution which generalizes. We avoid choosing the Sure Thing hypothesis by expressing an *a priori* preference for simpler solutions using the principle of Occam's Razor. MacKay and others have formalized this preference mechanism through the use of Bayesian methods. These provide a natural and quantitative embodiment of Occam's razor. The first level of inference requires that we find the best-fit parameters. The second level of inference requires the comparison of competing hypotheses. If we assume that the competing hypotheses are *a priori* equiprobable then the best hypothesis is chosen by evaluating the evidence. The evidence is computed by marginalization across the model parameters.

The evidence provides a natural trade-off between the best-fit likelihood and the Occam factor. This concept is closely related to other methods such as the Minimum Description Length and the Bayesian Information Criteria where the model is directly penalized by the number of parameters used. A similar idea was also incorporated into SVM models, which penalize the models with too large a capacity (VC dimension). However, while the SVM models are forced to estimate the penalty via cross-validation

schemes, Bayesian techniques automatically determine and apply the penalty in a fully probabilistic framework.

Approach	Error Rate	# Parameters
K-Nearest Neighbor	44%	
Gaussian Node Network	44%	
SVM: Polynomial Kernels	49%	
SVM: RBF Kernels	35%	83 SVs
Separable Mixture Models	30%	
RVM: RBF Kernels	30%	13 RVs

Table 3. Performance comparison of SVMs and RVMs to other nonlinear classifiers on static vowel classification data.

Assuming we have no prior knowledge that would cause us to favor a particular prior, we can find the optimal value for by evaluating the evidence. If we did have prior knowledge, we would simply repeat the inference over using the prior. At some level of the inference, we will arrive at a point where our prior knowledge is too weak to apply and then we evaluate the evidence. This extreme application of the Bayesian prior as a control parameter is known as the method of *automatic relevance determination* (ARD). It is so named because the prior over the input unit weights in a neural network can 'shut-off' those input dimensions which are irrelevant to the problem at hand. ARD is at the heart of the relevance vector machines that will be described next.

I.2 Relevance Vector Machines

An application of the evidence framework to kernel machines is the relevance vector machine (RVM). As with SVMs, RVMs use a weighted linear combination of basis functions. Due to the large number of parameters in this mode l— one per observation — we must guard against overfitting of the model to the training data. SVMs use a control parameter to implicitly balance the trade-off between training error and generalization. RVMs take a Bayesian approach and explicitly define an ARD prior distribution over the weights.

Each weight in the RVM model has an individual hyperparameter, , that is iteratively reestimated as part of the optimization process. As the hyperparameter grows larger, the prior on becomes infinitely peaked around zero, forcing to go to zero and, thus, contributing nothing to the summation. This process automatically embodies the principle of Occam's Razor because it explicitly seeks the simplest model that satisfies the data constraints. In practice, the majority of the weights are pruned, resulting in an exceedingly sparse model with generalization abilities on par with SVMs. To complete the Bayesian specification of the model, we have to specify a prior probability. In practice we use a non-informative (flat) prior to indicate a lack of preference.

The parameter estimation process is an iterative reduction process. That is, initially each vector of the system is allocated one parameter. As the procedure continues, vectors are pruned from the model when they are found to be irrelevant with respect to the remaining parameters. Integral to this iterative reestimation process is the computation of the inverse Hessian matrix. This operation requires the inversion of an MxM Hessian matrix where M is initially set to the size of training set. For larger training sets (on the order of a few thousand), this computation is prohibitive both in time and memory.

I.3 Experiments

RVMs have had significant success in several classification tasks. These tasks have, however, involved relatively small quantities of static data. Speech recognition, on the other hand, involves processing a very large amount of temporally evolving signals. In order to gain insight into the effectiveness of RVMs for speech recognition, we explored two tasks. We first experimented on the Deterding static vowel classification task which is a common benchmark used for new classifiers. Second, we applied the

techniques described above to a complete small vocabulary recognition task. Comparison with SVM models are given below. For each task, the RVMs outperformed the SVM models both in terms of model sparsity and error rate.

In our first pilot experiment, we applied SVMs and RVMs to a publicly available vowel classification task, Deterding Vowels. This was a good data set to evaluate the efficacy of static classifiers on speech classification data since it has been used as a standard benchmark for several nonlinear classifiers for several years. In this evaluation, the speech data was collected at a 10 kHz sampling rate and low pass filtered at 4.7 kHz. The signal was then transformed to 10 log-area parameters, giving a 10 dimensional input space. A window duration of 50 msec was used for generating the features. The training set consisted of 528 frames from eight speakers and the test set consisted of 462 frames from a different set of seven speakers. The speech data consisted of 11 vowels uttered by each speaker in a h*d context. Though it appears to be a simple task, the small training set and significant confusion in the vowel data make it a very challenging task.

Table 1 shows the results for a range of nonlinear classification schemes on the Deterding vowel data. From the table, the SVM and RVM are both superior to nearly all other techniques. The RVM achieves performance rivaling the best performance reported on this data (30% error rate) while exceeding the error performance of SVMs and the best neural network classifier. Importantly, the RVM classifiers achieve superior performance to the SVM classifiers while utilizing nearly an order of magnitude fewer parameters. While we do not expect the superior error performance to be typical (on pure classification tasks) we do expect the superior sparseness to be typical. This sparseness property is particularly important when attempting to build systems which are practical to train and test.

J. References

M. Beckman and J. Pierrehumbert, "Intonational structure in Japanese and English," Phonology Yearbook, 3, 255-309, 1986.

W. Byrne et al., "Pronunciation Modelling Using a Hand-Labelled Corpus for Conversational Speech Recognition," in Proc. ICASSP, 1998.

D. Bikel, On the Parameter Space of Lexicalized Statistical Parsing Models, Ph.D. thesis, University of Pennsylvania, 2004.

E. Charniak and M. Johnson, "Edit detection and parsing for transcribed speech," in Proc. NAACL 2001.

M. Gregory, M. Johnson, and E. Charniak, "Sentence-internal prosody does not help parsing the way punctuation does," in Proc. HLT-NAACL, 2004, pp. 81-88.

D. Hindle, "Deterministic parsing of syntactic non-fluencies," in Proc. ACL, 1983, pp. 123-128.

M. Johnson and E. Charniak, "A {TAG}-based noisy channel model of speech repairs," in Proc. ACL, 2004, pp. 33-39.

M. Johnson, E. Charniak and M. Lease, "An improved model for recognizing disfluencies n conversational speech," in Proc. NIST Rich Transcription Workshop, 2004, to appear.

J. Kahn, M. Ostendorf, and C. Chelba, "Parsing conversational speech using acoustic segmentation," in Proc. HLT-NAACL, comp. vol., 2004, pp. 125-128.

J. Kim, S. Schwarm and M. Ostendorf, "Detecting structural metadata with decision trees and transformation-based learning," in Proc. HLT-NAACL, pp. 137-144, May 2004.

Y. Liu, E. Shriberg, A. Stolcke and M. Harper, "Using machine learning to cope with imbalanced classes in natural speech: Evidence from sentence boundary and disfluency detection," in Proc. ICSLP, 2004.

Y. Liu et al., "Structural metadata research in the EARS program," in Proc. ICASSP, to appear, 2005.

L. Mayfield, M. Gavalda, Y. Seo, B. Suhm, W. Ward, and A. Waibel, "Parsing real input in {JANUS}: a concept-based approach," in Proc. TMI 95, 1995.

E. Noth, A. Batliner, A. Kiebling, R. Kompe, and H. Niemann, "Verbmobil: The Use of Prosody in the Linguistic Components of a Speech Understanding System," IEEE Trans. on Speech and Audio Processing, 8(5):519-532, 2000.

M. Ostendorf, "Linking Speech Recognition and Language Processing Through Prosody," CC-AI, vol. 15, no. 3, pp. 279-303, 1998.

M. Ostendorf, I. Shafran, S. Shattuck-Hufnagel, B. Byrne and L. Carmichael, "A prosodically labeled database of spontaneous speech," Proc. of the ISCA Workshop on Prosody in Speech Recognition and Understanding, pp. 119-121, October 2001.

J. Pitrelli, M. Beckman, and J. Hirschberg, "Evaluation of prosodic transcription labeling reliability in the ToBI framework," In Proc. of the International Conference on Spoken Language Processing, 1, 123-126, 1994.

P. J. Price, M. Ostendorf, S. Shattuck-Hufnagel and C. Fong, "The Use of Prosody in Syntactic Disambiguation," Journal of the Acoustical Society of America, vol. 90, no. 6, December 1991, pp. 2956-2970.

I. Shafran, M. Ostendorf, and R. Wright, "Prosody and phonetic variability: lessons learned from acoustic model clustering," Proc. of the ISCA Workshop on Prosody in Speech Recognition and Understanding, pp. 127-131, October 2001.

E. Shriberg et al., "Prosody-based automatic segmentation of speech into sentences and topics," Speech Communication, 32(1-2), pp. 127-154, 2000.

D. Talkin, "Pitch Tracking," in Speech Coding and Synthesis, ed. W. B. Kleijn and K. K. Paliwal, Elsevier Science B.V., 1995.

F. Jelinek, *Statistical Methods for Speech Recognition*, MIT Press, Cambridge, Massachusetts, USA, 1997.

F. Jelinek, "Up From Trigrams! The Struggle for Improved Language Models," *Proceedings of the 2nd European Conference on Speech Communication and Technology (Eurospeech)*, pp. 1037-1040, Genova, Italy, September 1991.

L. R. Rabiner, "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257-285, February 1989.

L. R. Bahl, P. F. Brown, P. V. de Souza, and R. L. Mercer, "Maximum mutual information estimation of hidden Markov model parameters for speech recognition," *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, pp. 49-52, Tokyo, Japan, October 1986.

P. Woodland and D. Povey, "Very Large Scale MMIE Training for Conversational Telephone Speech Recognition," *Proceedings of the 2000 Speech Transcription Workshop*, University of Maryland, MD, USA, May 2000.

S. Renals, *Speech and Neural Network Dynamics*, Ph. D. dissertation, University of Edinburgh, UK, 1990.

A. J. Robinson, *Dynamic Error Propagation Networks*, Ph.D. dissertation, Cambridge University, UK, February 1989.

J. Tebelskis, S*peech Recognition using Neural Networks*, Ph. D. dissertation, Carnegie Mellon University, Pittsburg, USA, 1995.

A. Ganapathiraju, *Support Vector Machines for Speech Recognition*, Ph.D. Dissertation, Mississippi State University, Mississippi State, Mississippi, USA, 2002.

M.D. Richard and R.P. Lippmann, "Neural Network Classifiers Estimate Bayesian a Posteriori Probabilities", *Neural Computation*, vol. 3, no. 4, pp. 461-483, 1991.

H.A. Bourlard and N. Morgan, *Connectionist Speech Recognition — A Hybrid Approach*, Kluwer Academic Publishers, Boston, USA, 1994.

G.D. Cook and A.J. Robinson, "The 1997 ABBOT System for the Transcription of Broadcast News," *Proceedings of the DARPA Broadcast News Transcription and Understanding Workshop*, Lansdowne, VA, USA, 1998.

V.N. Vapnik, *Statistical Learning Theory*, John Wiley, New York, NY, USA, 1998.

C. Cortes, V. Vapnik. Support Vector Networks, *Machine Learning*, vol. 20, pp. 293-297, 1995.

C.J.C. Burges, A Tutorial on Support Vector Machines for Pattern Recognition, http:// svm.research.bell-labs.com/SVMdoc.html, AT&T Bell Labs, November 1999.

B. Schölkopf, C. Burges and A. Smola, *Advances in Kernel Methods: Support Vector Machines*, MIT Press, Cambridge, MA, USA, December 1998.

O. L. Mangasarian, W. Nick Street and W. H. Wolberg: ‘‘Breast cancer diagnosis and prognosis via linear programming", *Operations Research*, 43(4), pp. 570-577, July-August 1995.

Y. Le Cun, L.D. Jackel, L. Bottou, C. Cortes, J. Denker, H. Drucker, I. Guyon, U.A. Muller, E. Sackinger, P. Simard, and V. Vapnik, "Learning algorithms for classification: A comparison on handwritten digit recognition," in *Neural Networks: The Statistical Mechanics Perspective*, J.H. Kwon and S. Cho, eds., pp. 261--276, World Scientific, 1995.

T. Joachims, "Text Categorization with Support Vector Machines: Learning with Many Relevant Features," *Technical Report 23, LS VIII, University of Dortmund*, Germany, 1997.

P. Clarkson and P. Moreno, "On the Use of Support Vector Machines for Phonetic Classification," *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, Phoenix, Arizona, USA, 1999.,

A. Ganapathiraju, J. Hamaker and J. Picone, "Support Vector Machines for Speech Recognition," *Proceedings of the International Conf. on Spoken Language Processing*, Sydney, Australia, Nov. 1998.

A. Ganapathiraju, J. Hamaker and J. Picone, "A Hybrid ASR System Using Support Vector Machines," submitted to the *International Conf. of Spoken Language Processing*, Beijing, China, October, 2000.

A. Ganapathiraju, J. Hamaker and J. Picone, "Hybrid HMM/SVM Architectures for Speech Recognition," *Proceedings of the Department of Defense Hub 5 Workshop*, College Park, Maryland, USA, May 2000.

M. Ostendorf, V. Digalakis, and O. Kimball, "From HMM's to Segment Models: A Unified View of Stochastic Modeling for Speech Recognition," *IEEE Transactions on Speech and Audio Processing*, vol. 4, no. 5, pp. 360-378, 1996.

M. Ostendorf and S. Roukos, "A Stochastic Segment Model for Phoneme-Based Continuous Speech Recognition," *IEEE Trans. on Acoustics Speech and Signal Processing*, vol. 37, pp. 1857- 1867, 1989.

M. Tipping, "Sparse Bayesian Learning and the Relevance Vector Machine," *Journal of Machine Learning*, vol. 1, pp. 211-244, June 2001.

M. Tipping, "The Relevance Vector Machine," in *Advances in Neural Information Processing Systems*, S. Solla, T. Leen and K.-R. Muller, eds., pp. 652-658, MIT Press, 2000.

D. J. C. MacKay, *Bayesian Methods for Adaptive Models*, Ph. D. thesis, California Institute of Technology, Pasadena, California, USA, 1991.

D. J. C. MacKay, "Probable networks and plausible predictions --- a review of practical Bayesian methods for supervised neural networks," *Network: Computation in Neural Systems*, 6, pp. 469-505, 1995.

E. T. Jaynes, "Bayesian Methods: General Background," *Maximum Entropy and Bayesian Methods in Applied Statistics*, J. H. Justice, ed., pp. 1-25, Cambridge University Press, Cambridge, UK, 1986.

S. F. Gull, "Bayesian Inductive Inference and Maximum Entropy," *Maximum Entropy and Bayesian Methods in Science and Engineering, vol. 1, Foundations*, G. J. Erickson and C. R. Smith, eds., Kluwer Publishing, 1988.

J. Rissanen, "Modeling by Shortest Data Description," *Automatica*, 14, pp. 465-471, 1978.

G. Schwartz, "Estimating the Dimension of a Model," *Annals of Statistics*, vol. 6, no. 2, pp. 461-464, 1978.

J. Hamaker, J. Picone, and A. Ganapathiraju, "A Sparse Modeling Approach to Speech Recognition Based on Relevance Vector Machines," *Proceedings of the International Conference of Spoken Language Processing*, vol. 2, pp. 1001-1004, Denver, Colorado, USA, September 2002.

E. Osuna, R. Freund, and F. Girosi, "Support Vector Machines: Training and Applications," *MIT AI Memo 1602*, March 1997.

A. Faul and M. Tipping, "Analysis of Sparse Bayesian Learning," *Proceedings of the Conference on Neural Information Processing Systems*, preprint, 2001.

J. Hamaker, *Sparse Bayesian Methods for Continuous Speech Recognition*, Ph.D. Dissertation, Mississippi State University, Mississippi State, Mississippi, USA, 2003.

M. E. Tipping and A. Faul, "Fast Marginal Likelihood Maximisation for Sparse Bayesian Models," submitted to *Artificial Intelligence and Statistics '03*, 2003.

D. Deterding, M. Niranjan and A. J. Robinson, "Vowel Recognition (Deterding data)," Available at *http://www.ics.uci.edu/pub/machine-learning-databases/undocumented/connectionist-bench/vowel*, *2000*.

P. Loizou and A. Spanias, "High-Performance Alphabet Recognition," *IEEE Transactions on Speech and Audio Processing*, pp. 430-445, 1996.

R. Cole, "Alphadigit Corpus v1.0," *http://www.cse.ogi.edu/CSLU/corpora/alphadigit*, Center for Spoken Language Understanding, Oregon Graduate Institute, USA, 1998.

J. Hamaker and J. Picone, "Iterative Refinement of Relevance Vector Machines for Speech Recognition," submitted to *Eurospeech'03*, Geneva, Switzerland, September 2003.