## 08/15/03 — 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 $\Theta$ 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($\Theta$) 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

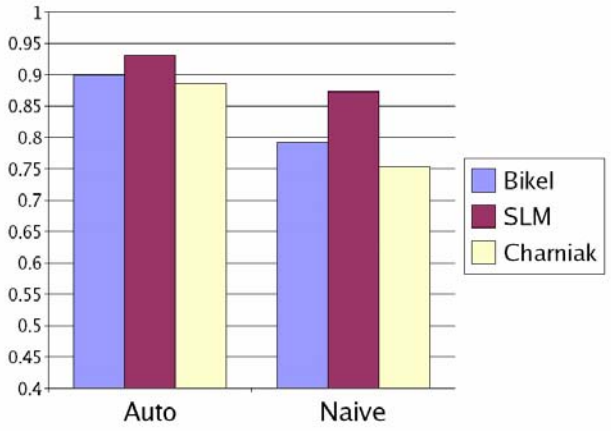


Figure 1. Relative performance of different parsers for the RT03 MDE SU detection system and a simple pause-based system, in each case comparing the F-measure to that obtained with oracle SUs.

[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.

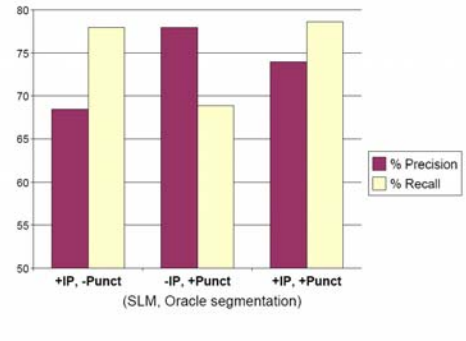


Figure 2. Precision-recall trade-offs associated with using punctuation, oracle IPs or both in addition to SUs in parsing.

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 Experiments

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.

## 08/15/02 — 08/14/03: RESEARCH AND EDUCATIONAL ACTIVITIES

In the third year of this project, we focused our efforts in four areas:

- **Interaction of Speech and Parsing**: investigated the use of prosody information within the framework of probabilistic parsing; developed new approaches to discriminative learning for sequence segmentation problems; investigated text categorization based on automatically extracted semantic features.
- **Language Modeling Technology**: developed neural network approaches to probabilistic language models;
- **Prosody and Parsing**: analyzed spectral cues to prosodic structure and the interaction between prosodic and syntactic structure.
- **Relevance Vector Machines**: focused on the creation of algorithms for kernel-based discriminative modeling that scale up to very large data sets.

These developments are described in more detail in the sections below. A project meeting will be held during Summer'03 to coordinate the work on this project.

### A. Laboratory for Linguistic Information Processing, Brown University

This year, our primary focus was the investigation of the use of prosody information within the framework of probabilistic parsing. We treated prosody much like punctuation in written text, i.e. as if it were separate "words." Unfortunately, as opposed to punctuation, we were not able to obtain any benefit from the prosody markers. The results were robust over a large number of trials in which the specifics of the markings were varied.

In the information retrieval component of our research, we explored two research problems: (i) discriminative learning for sequence segmentation problems and label sequence problems and (ii) robust text categorization based on automatically extracted semantic features.

Learning from observation sequences is a fundamental problem in machine learning. One facet of the problem generalizes supervised classification by predicting label sequences instead of individual class labels. The latter is also known as label sequence learning. It subsumes problems like segmenting observation sequences and annotating observation sequences. The potential applications are widespread, most interesting in the context of the current project is the application for named entity classification and information extraction to support information retrieval. We have made two major contributions to this problem. First, we have developed a boosting algorithm for label sequence learning and investigated the use of several alternative objectives for the label sequence learning problem [1]. Secondly, we have proposed a novel algorithm that combines Support Vector Machine learning with Hidden Markov Models [2,3], resulting in a powerful learning architecture that combines the high classification accuracy of large margin methods, with the flexibility of kernels, and the efficiency of dynamic programming. Experimental comparisons for named entity classification and part-of-speech tagging have proved the competitiveness of our approach compared to state-of-the-art techniques like conditional random fields.

In order to improve the accuracy and robustness of term based text categorization we have investigated the use of semantic features in addition to simple term based features [4]. The semantic features are domain-specific and are automatically extracted from a document collection without the need for a thesaurus or any other linguistic resource. We have employed a technique

called probabilistic Latent Semantic Analysis (pLSA) to that extend. Experimental results show consistent and significant improvements with respect to various performance measures (such as accuracy, average precision, precision-recall breakeven, etc.) compared to purely term-based methods.

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

The biggest obstacle language modeling research has to overcome is the 'curse of dimensionality.' If we want to have a model which is capable of assigning scores (probabilities) to long sequences of words, given vocabulary sizes on the order of tens of thousands of words, the model will have far too many parameters to be reliably estimated even with the largest corpora available. Our approach is two-fold: (1) transforming the problem into a smooth continuous domain where any seen event in the training corpus will contribute to the estimation of the probabilities of its unseen neighbors, (2) developing a combinatory categorical grammar (CCG) as a natural enrichment of the syntactical labels in the structured language model (SLM). We briefly describe progress in each of these areas.

N-gram language models are the most commonly used models in speech recognition systems. Despite their naive underlying assumption, N-gram models perform surprisingly well. However, they suffer from severe data sparseness, and are intrinsically unable to use long contexts for prediction. In the SLM [11], long contexts are used for prediction by means of building partial syntactical parses on the prefix word strings and using information extracted from these partial parses.

There has been promising work in using distributional representations of words and neural networks for language modeling [12]. One great advantage of this approach is its ability to fight data sparseness. The model size grows only sub-linearly with the number of predicting features used. The SLM is made of three components: a predictor which predicts the next word, a tagger which tags the newly predicted word, and a constructor which builds partial parses for the newly extended word string. The neural network approach, on the other hand, uses a feature vector which is associated with each token in some given input vocabulary. The input to the network is a single vector that is a concatenation of the feature vectors of the items in the history. The neural network then computes the (conditional) distribution over all tokens in the output vocabulary given the input described above.

The neural network approach has been successful at reducing the perplexity on the UPENN section of the Wall Street Journal task from 132 (our previous baseline SLM result) to 117. The latter result uses three previous head plus the first opposite head [13] along with a 5-gram backoff model. We have also investigated re-ranking the output of a speech recognizer on the WSJ task using the neural network-based SLM (NN-SLM). WER was reduced from 13.2 to 12.4 by interpolating a lattice output and rescoring using the 5-gram based SLM. We believe these preliminary results are promising, and can be improved by full embedded training of the NN-SLM.

The combinatory categorical grammar (CCG) is a wide-coverage parsing technique that has the potential benefit of a more constrained grammar and simple semantically transparent capture of extraction and coordination [14]. CCG grammars have much larger category sets than standard Penn Treebank grammars that we used in our previous SLM studies [15,16]. For example, CCGs distinguish between many classes of verbs with different subcategorization frames. As a result of

simple unary and binary combinatory schemata such as function application and composition, CCG has a smaller and less overgenerating grammar than standard PCFGs.

Our interest in using CCGs in SLM lies in the fact that CCG categories can serve as a natural enrichment of the syntactic information of a lexical item or a constituent in the parse tree. Since CCG categories are context dependent, we will have a context dependent enrichment of the syntactic heads, as opposed to uniform enrichment in previous studies. We investigated the perplexity performance of the CCG style SLM on the UPenn Treebank data. The CCG-SLM reduced perplexity from 166 to 147, a 15.1% reduction. We then investigated its impact on N-best rescoring of a WSJ task and found it provided no significant reduction in WER.

There are many issues that need further investigation to accurately assess the CCG-SLM performance on a recognition task. For example, we currently use linear interpolation as the smoothing method in all models. This is not likely to be the optimal choice for the CCG-SLM because of the large number of categories we are using. We plan to investigate other alternatives for smoothing such as history clustering and maximum entropy.

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

This year, we used prosodically labeled data as a feature in HMM decision tree clustering to investigate whether prosody might be useful in acoustic modeling and whether spectral cues might be useful for prosody recognition. Preliminary results suggest that spectral cues are useful for discriminating between fluent and disfluent pauses.

We also analyzed the interaction between prosodic and syntactic structure. We investigated different aspects of parse structure to determine what features are the best predictors of prosodic structure, with the assumption that these would be the best targets for using prosody to improve parsing. The depth of the left constituent is the most important feature of those investigated.

We continued investigating recognition of prosodic structure given acoustic cues and/or syntactic cues using known word transcriptions and simple decision tree classifiers. For a 4-class recognition problem, we achieved 79% correct with prosodic cues alone, and 89% correct when parse features are added. Since very high results are obtained with the combined features, and there is much more syntactically annotated data than prosodically labeled data, we are currently investigating training prosody recognition modules using only partially labeled data.

Finally, we experimented with recognition of sentence boundaries and disfluency interruption points. We have begun experiments in recognizing sentence boundaries, incomplete sentences and disfluency interruption points using prosodic and word class (POS) cues. So far, we have good results for detecting sentences, but interruptions and incomplete sentences are much less frequent and hence not well modeled. One problem is that there are word boundaries that could (theoretically and because of acoustic correlates) be marked as interruption points (e.g. before a filled pause) which were not marked with the current labeling convention. The next step is to assess performance with a revised disfluency labeling system.

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

The work at Mississippi State University this year has focused on the creation of algorithms for kernel-based discriminative modeling that scale up to very large data sets. At the close of the last fiscal year, we had developed a hybrid relevance vector machine (RVM) recognition system

which addressed the major limitations in our hybrid support vector machine (SVM) system described in [5,6]. The Relevance Vector Machine (RVM) [7] attempts to overcome the deficiencies of the SVM by incorporating a probabilistic model directly into the classifier rather than using a large margin classifier [7]. The principle attraction of the RVM is that it delivers comparable performance as an SVM, but uses much fewer parameters. It is also much more computationally efficient during testing as described in Table 1.

As with SVMs, the process to train an RVM classifier is computationally expensive even for small problems. For the RVM, the training procedure uses 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. In fact, initially in this work we were unable

to operate on data sets larger than a few thousand training examples.

This year, we have focused on methods to overcome the training size limitation with the RVM. Our first attempt was to implement a constructive training approach recently defined by Tipping and Faul [8]. In this algorithm the model begins with only a single parameter specified. All others are implicitly pruned. Parameters are then added to the system in a constructive fashion while still satisfying the original optimization function. We are able to add a good bit more training data to our system — on the order of 10 thousand examples — in training. However, care must be taken to insure convergence rates are reasonable. We have found that the model will often oscillate between a few local optima leading to slow convergence or even an inability to converge.

Despite our ability to increase the overall training size by approximately one order of magnitude, this iterative procedure does not completely solve the problem. For even larger problems as are typical in speech recognition, the full design matrix (or kernel matrix), will not fit in memory. We can still use the constructive approach but it requires the repeated recalculation of the full design matrix and is, again, prohibitive — now in time rather than memory. We have spent considerable effort this fiscal year researching alternatives to the constructive approach of Tipping and Faul that can overcome this problem. We have developed automatic data selection methods that allow one to determine which training vectors are most likely to contribute to the final model.

Each of the data selection methods follow the same essential algorithm demonstrated in Figure 1.

| 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 Alphadigit recognition data. The RVMs yield a large reduction in the parameter count while attaining superior performance. However, this performance comes with a large up-front cost in training.

A seed model is trained from a small (but reasonably-sized) data set. This model is then used to probe the remaining training vectors. Some measure is used to determine which of the remaining candidate training vectors are most appropriate to add to the training set. Training is then repeated with the candidate vectors added to the model pool. Either the constructive training approach or the iterative pruning approach to training can be used at this point so long as the model remains suitably small.

What differentiates the data selection methods we have examined is the criteria used to measure the "goodness" of the candidate vectors. Our first selection criteria is drawn from Faul and Tipping [9]. The marginal log-likelihood of the data can be written as:

$$L(\alpha) = -\frac{1}{2}[N\log 2\pi + \log|\boldsymbol{C}| + \boldsymbol{t}^T\boldsymbol{C}^{-1}\boldsymbol{t}] \quad (1)$$

where $\boldsymbol{C}$ is defined in terms of the model parameters and $\boldsymbol{t}$ are the training targets of the system. Tipping and Faul define the incremental, predicted, change in the log-likelihood function due to the addition of a training parameter. Our method uses that incremental change as the selection criterion. Those vectors that have a maximal predicted change, given the current model, are chosen to be candidates for the next model.

The second selection criteria used follows the work of MacKay [10]. The mean marginal

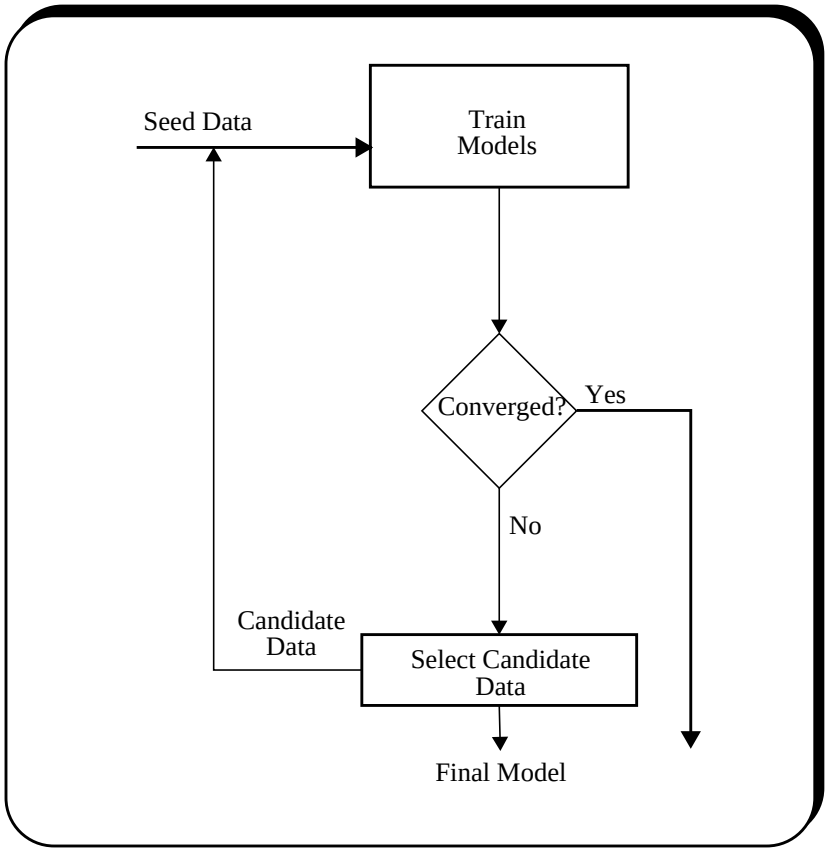


Figure 1. Flow graph for data selection methods. The criteria for candidate selection differentiates the different methods.

information gain of $t_i$ at point $x_i$ when $x_i$ is added to the model is given by

$$\frac{1}{2}\Delta\log\sigma_x^2 \tag{2}$$

where $\sigma_x^2$ can be computed directly from the current model parameters. The goal, then, would be to find those vectors which maximize (2). There is an analogous, but more complex, function for choosing a set of data that produces a joint maximum. These techniques have each increased the training capacity of the RVM approach by two orders of magnitude. We are still in the process of evaluating the properties of these. Particularly, we are interested in their convergence properties and their effect on word error rate in the recognizer.

## E. REFERENCES

[1] Yasemin Altun, Thomas Hofmann & Mark Johnson, Discriminative Learning for Label Sequences via Boosting, Advances in Neural Information Processing Systems (NIPS*15), 2003, to appear

[2] Thomas Hofmann and Yasemin Altun, Large Margin Methods for Label Sequence Learning, Proceedings 8th European Conference on Speech Communication Technology (EUROSPEECH), 2003, accepted for publication

[3] Yasemin Altun, Ioannis Tsochantaridis and Thomas Hofmann, Hidden Markov Support Vector Machines, Submitted to International Conference on Machine Learning (ICML), 2003.

[4] Lijuan Cai & Thomas Hofmann, Text Categorization by Boosting Automatically Extracted Concepts, Submitted to ACM Information Retrieval Conference (SIGIR), 2003

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

[6] 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.

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

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

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

[10] D. J. C. MacKay, "Information-based objective functions for active data selection," *Neural Computation*, vol. 4, no. 4, pp. 590-604, 1992.

[11] A. Emami, P. Xu and F. Jelinek, "Using a connectionist model in a syntactical based language model, to be presented at the International Conference on Acoustics, Speech, and Signal Processing, Hong Kong, April 2003.

[12] A. Emami, "Improving a connectionist based syntactical language model," submitted to the European Conference on Speech Technology, Geneva, Switzerland, September 2003.

[13] A. Emami and F. Jelinek, "Neural Probabilistic Structured Language Modeling," Technical Report 07/02-12/02, Center for Language and Speech Processing, Johns Hopkins University, Baltimore, Maryland, USA, December 2002.

[14] J. Hockenmaier and M. Steedman, "Generative models for statistical parsing with Combinatory Categorical Grammar," in Proceedings of the 40th Annual Meeting of the ACL, pp. 335-342, Philadelphia, Pennsylvania, USA, July 2002.

[15] C. Chelba and F. Jelinek, "Structured Language Modeling," in Computer, Speech, and Language, 14(4):283-232, October 2000.

[16] P. Xu, C. Chelba, and F. Jelinek, "A study on richer syntactic dependencies for structured language modeling," in Proceedings of the 40th Annual Meeting of the ACL, pp. 335-342, Philadelphia, Pennsylvania, USA, July 2002.

## 08/15/01 — 08/14/02: RESEARCH AND EDUCATIONAL ACTIVITIES

In the second year of this project, we focused our efforts in four areas:

- **Interaction of Speech and Parsing**: investigated the way in which language phenomena which are very common in speech, but relatively rare in the formal text that parsing technology typically deals with, effects the parsing process.
- **Lattice Generation Technology**: developed lattice cutting techniques that transforms traditional word lattices into a series of segment sets that contain confusable words and phrases, thereby simplifying the search process during rescoring.
- **Prosody and Parsing**: developed categorical prosodic break labels, building on linguistic notions of minor and major prosodic phrases and the hesitation phenomena.
- **Relevance Vector Machines**: developed new reestimation techniques to make this approach feasible for large-scale system evaluations.

These developments are described in more detail in the sections below.

A project meeting was held at Brown University on June 13, 2002 to coordinate the work on this project. All organizations involved in this project were present at this meeting. Our next joint project meeting is planned for early June'2003.

### A. Laboratory for Linguistic Information Processing, Brown University

One of the research activities this last year on the interaction between speech and parsing was an investigation into the way in which language phenomena which are very common in speech, but relatively rare in the formal text that parsing technology typically deals with, effects the parsing process. In particular, both "filled pauses" ("ums" and "ahs") and parentheticals ("you know") are common in speech, but not text. Previous work in the area has shown that both tend to occur more readily at clause boundaries than elsewhere in sentences, leading to the conjecture that rather than making parsing more difficult, they might make things easier. Unfortunately, some recent experiments at Brown, to be presented at the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP) seem to suggest that this is not the case. A standard statistical parser was trained on text with and without such phenomena, and its performance was measured. It seems that filled pauses make parsing harder, parentheticals make parsing harder, and both together make it harder still. This goes against the prevailing expectations, and some recent suggestions by the prosody researchers within this project are going to be followed up upon in an effort to refine this result. In particular there is some evidence that "ah" and "um" might behave differently in this regard, and it might be worth distinguishing the two, something not done in this last year's experiments.

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

CLSP concentrated on three activities this year: lattice cutting, neural probabilistic language modeling, and the impact of richer syntactic dependencies on the performance of the structured language model.

### B.1 Lattice Cutting

CLSP has provided lattices for use in developing parsers automatically transcribe speech. Lattices were generated on the RT-02 (Rich Transcription 2002 Evaluation) development test set using a

conversational speech ASR system trained on the SWITCHBOARD corpus. These were provided in determinized acoustic score form, in that the language model scores used in generating the lattices were removed so that only the acoustic score of each individual word hypotheses remained. These were determinized using the AT&T FSM toolkit so that each lattice is compact and easy to search using any left-to-right language model.

A segmented version of these lattices were also produced using lattice cutting techniques developed at CLSP. Lattice cutting pinches word lattices so that lattices are transformed to look like a series of segment sets that confusable words and phrases. Lattice rescoring is changed from searching over entire sentences found in the original lattice to resolving the small number of confused words and phrases in the segment sets. A research avenue to explore is whether parsers can be modified to search over these smaller, more constrained sets.

## B.2 Neural Probabilistic Language Modeling

The problem of language modeling research for ASR is essentially the problem of sparseness of data. Conventionally, it has been treated by smoothing of various kinds and lately by utilization of sentence structure. However, Bengio and his associates [1] have come up with a novel approach based on artificial neural networks (ANNs). We have confirmed their results by independent experimentation, as shown in Table 1.

Perplexity experiments were carried out on the Brown corpus and the UPenn section of the WSJ corpus. We will next (a) ascertain the ASR error rate effects obtainable from these improvements (b) apply the approach to improving components (i.e., the predictor) of a structural language model.

## B.3 The Impact Richer Syntax Dependencies on the Structured Language Model

We studied the impact of richer syntactic dependencies on the performance of the structured language model (SLM) along two dimensions: perplexity (PPL) and word-error-rate (WER, N-best rescoring).

Under the equivalence classification in the SLM, the conditional information available to the SLM model components is made up of the two most-recent exposed heads consisting of two NT tags and two headwords. In an attempt to extend the syntactic dependencies beyond this level, we enriched the non-terminal tag of a node in the binarized parse tree with the NT tag of the parent node (PA), or the NT tag of the child node from which the headword is not being percolated (OP), or we added the NT tag of the third most-recent exposed head to the history of the CONSTRUCTOR component (h-2).

Without interpolating with the 3-gram, the opposite (OP) scheme performed the best, reducing the PPL of the baseline SLM by almost 5%

| Corpus Subset | Baseline | Neural Network | Combined |
|---|---|---|---|
| Brown | 366 | 257 | N/A |
| UPenn | 141 | 157 | 121 |

Table 1. A comparison of a neural network (NN) based language modeling technique to traditional methods. The baselines 3-gram interpolated and 5-gram Knesser-Ney interpolated respectively. For the combined case, the NN model was interpolated with the baseline using a constant weight of 0.5023.

relative. When the SLM is interpolated with the 3-gram, the h-2+opposite+parent scheme performed the best, reducing the PPL of the baseline SLM by 3.2%.

The h-2+opposite scheme achieved the best WER result, with a 0.4% absolute reduction over the performance of the opposite scheme. Overall, the enriched SLM achieves 10% relative reduction in WER over the 3-gram model baseline result. This scheme outperformed the 3-gram used to generate the lattices and N-best lists, without interpolating it with the 3-gram model.

We will continue to study additional changes in the SLM parametrization schemes.

## 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 [2]. While there is evidence that both phrase-level emphasis (or, prominence) and prosodic phrases provide information for syntactic disambiguation [7], the most important cue seems to be phrase structure. The 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.

### 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, substantial samples of the Switchboard corpus of spontaneous telephone quality dialogs were labeled using a simplification of the ToBI system for transcribing pitch accents, boundary tones and prosodic constituent structure [6]. 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.isip.msstate.edu/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) [3]. All conversations were analyzed using a high quality pitch tracker [9] 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 labeller uncertainty about a particular transcription. The annotation does not include accent tones, 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.

## C.2 PROSODY AND PARSING

Our approach is to integrating prosody in parsing is to use categorical prosodic break labels, building on linguistic notions of minor and major prosodic phrases and the hesitation phenomena. An important reason for using categorical units rather than the acoustic correlates themselves is that the intermediate representation simplifies training with high-level (sparse) structures. Just as most 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 prosodic breaks are also useful as a mechanism for generalizing the large number of continuous-valued acoustic features to different parse structures. In addition, the low-dimensional discrete representation is well suited to integration with current parsing frameworks, either as added “words” or as features on words. This approach is currently somewhat controversial because of the high cost of prosodic labeling, and to some extent because of the association with a particular linguistic theory. The the specific subset of labels used in this work, though founded in the ToBI system, collapse some of the detail of the system to simply represent minor and major phrases and disfluencies, so in fact the categories are relatively theory neutral (and language independent). Furthermore, a key objective of this work is to overcome the cost of prosodic labeling by using bootstrapping techniques.

Specifically, a small set of labeled data is used to train an automatic prosody annotation algorithm that has both text and acoustic cues. These cues are used in combination to automatically label the rest of the Switchboard data, and then new (separate) prosody-parse and prosody-acoustic models are designed for the final system, building on EM or co-training techniques. An alternative approach, as in [4], 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 Switchboard because of the longer utterance and potential confusion between fluent and disfluent pauses.

The automatic prosody annotation effort is described further in the next section; here we briefly outline the key research issues for designing and integrating the different model components for the parsing application. Building on the two-stage approach introduced in [10], the planned architecture involves prosodic break detection and generation of an augmented word transcription, followed by detection of edit points and disfluent regions, and finally parsing. Key research issues include whether the prosodic breaks should be treated as “words” or as features on words, whether disfluencies should be represented as an independent component, and how to represent uncertainty of the prosodic classifier.

## C.3 AUTOMATIC LABELING OF PROSODIC STRUCTURE

An important part of the past year’s effort was development of a prosodic labeling system in order to increase the effective training data, and for analysis of the dependence between prosodic and

parse structures in conversational speech. Experiments were conducted on the Switchboard corpus, specifically the prosodically annotated subset described previously. We used decision tree classifiers with a combination of acoustic, punctuation, parse, and disfluency cues. In this initial study, the only acoustic cue was silence duration; work with F0, energy and duration cues is in progress. The punctuation, parse and disfluency cues were taken from the annotations available from the Linguistic Data Consortium, aligned to the word transcriptions as described above. Speaker turn and incomplete sentence markers were among these cues. The disfluency file included repair points as well as markers of filled pauses and coordinating conjunctions used in utterance initial position. 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.

The baseline system assigns the most likely class (default word boundary) in all cases except for assigning a major phrase boundary at pause locations. This strategy gives 74% accuracy. Using disfluency, parse features and silence duration features improves performance to 86% accuracy. The most important features are punctuation, silence duration, disfluency edit point markers, left-to-right depth of the parse tree, and part-of-speech tags. The tree was quite simple (8 nodes). We anticipate further performance gains with the use of duration lengthening and intonation cues.

From analysis of the resulting tree, we find that punctuation and repair points are correlated with prosodic breaks and hesitations, as expected. Silences alone (though useful) are 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. The depth (or possibly the length) of the left constituent is a useful predictor, but labels of the neighboring constituents do not. Further investigation of representations of syntactic constituent labels is ongoing, since other studies have shown association of clause boundaries (e.g. constituent labels) with major prosodic breaks. We find that minor phrase boundaries are never predicted by the decision trees designed in these experiments. Although this is not entirely surprising, since minor phrases are rare, we think that it may be possible to distinguish these structures given duration lengthening and/or intonation cues, in which case analysis of the full Switchboard corpus could show some relationship between minor phrases and particular syntactic 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

The work at Mississippi State University this year has centered on extending last year’s progress in acoustic modeling robustness through kernel-based discriminative modeling. At the close of the last fiscal year, we had developed a hybrid speech recognition system that combined the temporal modeling benefits of hidden Markov models (HMMs) and the discriminative modeling capabilities of the support vector machine (SVM) paradigm. This hybrid system used a segmental modeling approach to phone classification, building a set of one-vs-all binary classifiers. To integrate the SVM into the HMM framework, a sigmoidal posterior probability function was used to convert the SVM distances to probabilities. From this work, we identified two major limitations of the hybrid HMM/SVM framework that have become the core of this year’s work:

- **HMM-derived segmentations:** In the hybrid system, the SVM is dependent on the HMM core to provide good segmentations. It would be preferable to have the SVM determine for itself an optimal segmentation and hypothesis set.
- **Ad-hoc probability estimator:** The sigmoid posterior estimate incorporated into the hybrid HMM/SVM system was found to be ineffectual — a follow-up experiment indicated that simply using a step function yielded only a negligible loss in accuracy. The relevance vector machine (RVM), a completely probabilistic model that retains many of the discrimination and sparsity properties of the SVM, was identified as a potential solution to this problem.

An initial method for removing the dependency of the SVM on the HMM segmentations was built upon a time-synchronous Viterbi decoder. The SVM in this system is presented with all possible phone segmentations for all possible hypotheses. It scores those according to the one-vs-all binary classifiers, and the search process chooses the best sequence of words given those scores. However, in this framework, we found that the computational resources required were too large. To achieve a reasonable resource requirement, pruning thresholds needed to be tuned to the point where overpruning frequently occurred. This resulted in search errors and very poor word error rates. For instance, on an alphadigits task where state-of-the-art error rates are in the range of 10-15%, the SVM system could only achieve 85% error.

An analysis of the search paths at runtime indicated that the problem was in the combined use of synchronous Viterbi search and segmental models. The segmental models require that a complete phone segment be hypothesized before the phone is actually hypothesized and scored. This results in a large number of hypotheses that exist in the same model at the same time but which can not be compared for pruning purposes. In other words, Viterbi pruning can not be carried out at the sub-phone level. Contrast this to standard HMM systems where the predominate pruning is the Viterbi pruning carried out a the sub-phone level. A potential solution to this problem is the implementation of a stack-based decoding approach. With the removal of the time-synchrony

| Classifier | Error Rate | Average Parameter Count |
|---|---|---|
| SVM | 35.0% | 82.8 |
| RVM | 30.3% | 12.6 |

Table 2. Comparison of SVM and RVM classifiers on Deterding vowel data. Each classifier type was trained as a set of 11 1-vs-all classifiers. The best performance reported thus far on this data is 30.4% using a speaker adaptation scheme called Separable Mixture Models.

| Classifier | Error Rate | Average Param. Count | Train Time | Test Time |
|---|---|---|---|---|
| SVM | 16.4% | 257 | 1/2 hour | 30 min |
| RVM | 16.2% | 12 | 1 month | 1 min |

Table 3. Comparison of SVM and RVM classifiers on alphadigit recognition tasks. Both systems used a segmental hybrid architecture. Note that the RVM has over an order of magnitude fewer parameters but requires significantly longer to train. A reduced training set of 2000 segments was used.

limitation, phone hypotheses can be pursued and pruned without the accumulation of many non-viable hypotheses.

A second line of research pursued this year was replacement of the SVM by an RVM model. The RVM is a Bayesian model which takes the same form as the SVM model and provides a fully probabilistic alternative to the SVMs which use the ad-hoc sigmoid posterior estimate. The RVMs have been found to provide generalization performance on par with SVMs while typically using nearly an order of magnitude fewer parameters as indicated for a vowel classification task in Table 2. Sparseness of the model is automatic using MacKay's automatic relevance determination methods.

Our initial attempts to incorporate the RVM technology used an approach identical to the hybrid HMM/SVM system. A set of one-vs-all RVM phone classifiers were trained on segmental data. Unlike the SVM, there was no need for a posterior estimator function since the RVM is, itself, a posterior estimator. As with SVMs, the process to train an RVM classifier is computationally expensive even for small problems. For the RVM, though, the computational complexity is $O(M^3)$ in run-time and $O(M^2)$ memory, where M is the number of basis functions and is initially set to the size of the training corpus. Thus, our initial attempts were limited to relatively small training sets as indicated in Tables 3 and 4. Since our aim is to replace the HMM emission distribution by an RVM, the RVM would be exposed to every frame of

| Classifier | Error Rate | Average Parameter Count |
|---|---|---|
| SVM | 40.8% | 1213 |
| RVM | 41.2% | 178 |

Table 4.. Comparison of SVM and RVM classifiers on alphadigit recognition tasks. Both systems used a segmental hybrid architecture. Note that the RVM has over an order of magnitude fewer parameters but requires significantly longer to train. A reduced training set of 2000 segments was used.

data in the training corpus. For even small speech corpora the number of frames in the training set is on the order of $10^6$. The usual RVM training methods are, thus, rendered impractical.

We are currently researching a number of alternative training schemes. Most of these incorporate a reduced set methodology where the optimization problem is solved for a small portion of the training data. The solution for that portion is then used to select other interesting portions of the training set that need to be examined. Eventually, an optimum on the entire training set is achieved through the optimization of many smaller sets.

The results of the HMM/SVM hybrid system indicate a need to automatically incorporate segmentation variation into the training process. HMMs offer a principled approach to this problem via the EM-based Baum-Welch algorithm. Our continued research aims to create a similar algorithm for training HMM/RVM systems. The RVM will replace the Gaussian as the frame-level emission distribution in the HMM state. Iterative reestimation formulae which describe cycles of Baum-Welch statistical accumulation (the expectation step) followed by Bayesian RVM training (maximization step) will be derived. In building this training algorithm we must address issues of iterative and monotonic convergence and stopping criteria. Similar work that has been developed for connectionist HMM/ANN systems will serve as reference.

## 08/15/00 — 08/14/01: RESEARCH AND EDUCATIONAL ACTIVITIES

In the first year of this project, we focused our efforts in two core areas:

- **Parsing Technology**: intimately coupling parsing technology with speech recognition technology and evaluating performance on conversational speech.
- **Risk Minimization in Acoustic Modeling**: developed a new acoustical modeling framework based on the principle of risk minimization using relevance vector machines; developed baseline recognition results for a related approach based on support vector machines.

We also began work on the integration of prosodic information into speech recognition and parsing. We developed a format for interfacing prosody output with our parser. We reviewed and cleaned up transcriptions of a prosodically labeled subset of the Switchboard corpus. We also discussed strategies for incorporating prosody into the search process in speech recognition.

A project kickoff meeting was held at Johns Hopkins University in June to coordinate the work on this project. All organizations involved in this project were present at this meeting. Discussions focused on three major topics: parsing, integration of prosody, and the development of resources to support this research. Plans were developed to begin evaluating the impact of parsing using a lattice rescoring approach, and to investigate the resources required to develop a time-aligned version of the Penn Treebank corpus that will be used for prosodic modeling. Other topics of discussion included some preliminary results on a hybrid speech recognition system using Support Vector Machines. Our next joint project meeting is planned for early June'2002.

### E. Parsing Technology

We have begun research into applying parsing technology to speech. While our ultimate goal is to intimately couple parsing technology with speech recognition technology, clearly a first step is to demonstrate that current parsing technology is in fact compatible with the kind of language that occurs in naturally-occurring speech, and demonstrating that current parsing technology can do a reasonable job of parsing speech transcripts is an important first step. State-of-the-art statistical parsers are invariably trained on Treebank training data, and the recent release of a treebanked portion of the Switchboard corpus by the LDC permitted us to train such a parser on spoken language transcripts. We have two papers that have already appeared in prestigious conferences, and one new result which we expect to submit to a 2002 conference. Charniak and Johnson [10] investigated the performance of state-of-the-art parser technology when applied to speech transcripts. Current parsing technology has been primarily developed using written material; indeed, the best high-performance statistical parsers available today are based on Wall Street Journal newspaper texts, and it was an open question whether this technology is applicable to spoken language.

Transcribed speech differs from edited written text in that it contains disfluencies of various kinds. The two major types of disfluencies we considered in this work are interjections (e.g., "ugh"), parentheticals (e.g., "Sam is, I think, insane") and speech repairs (e.g., "I told my brother, ugh, my sister I'd be late"). Interjections are extremely easy to recognize using standard part-of-speech tagging techniques, and there has been speculation in the literature that interjections provide valuable clues to phrase boundaries (we describe empirical an evaluation of this hypothesis below). Written text also contains parentheticals, and these do not seem to cause current parsing technology any particular problems. However, in a pilot experiment we determined our standard

statistical parser, even when trained from a Switchboard speech transcript treebank that identifies speech repairs, fails to identify any speech repairs in the test corpus. This is not too surprising, since modern statistical parsers function by modeling the tree-structured head-to-head dependencies in a normal natural language sentence, but speech repairs do not seem to be included in such dependencies. Charniak and Johnson [10] present a simple architecture for parsing transcribed speech in which an edited-word detector first removes such words from the sentence string, and then a standard statistical parser trained on transcribed speech parses the remaining words. The edit detector achieves a misclassification rate on edited words of 2.2%. (The **NULL**-model, which marks everything as not edited, has an error rate of 5.9 %.) To evaluate our parsing results we introduce a new evaluation metric, the purpose of which is to make evaluation of a parse tree relatively indifferent to the exact tree position of **EDITED** nodes. By this metric the parser achieves 85.3\% precision and 86.5\% recall; results which are comparable with the best written text parsing results of just a few years ago.

In [11], we investigated the use of our parsing model as a language model. Language models, of course, are used in speech recognition systems to distinguish between likely and unlikely word strings proposed by the speech recognizer's acoustic model. Most speech recognition systems use the very simple trigram language model, but recently there has been increased interest in using parsing for this task. However, the previous parsers used for this purpose have not performed parsing tasks at state-of-the-art levels. This is because the researchers assumed that any language model would have to work in a strict left-to-right fashion. Unfortunately, the best statistical parsers are "immediate-head" parser — our name for a parser that conditions all events below a constituent *c* upon the head of *c*. Because the head of a constituent may appear in the middle or at the end (e.g., the head of a noun-phrase is typically the right-most noun) immediate head parsers cannot work in a strict left-to-right fashion. However the reasons for preferring strict-left-to-right are not iron-clad and we were interested in determining if better parsing performance of immediate-head parsers would lead to a better language model. In the paper we presented two immediate-head language models. The perplexity for both of these models significantly improve upon the trigram model base-line as well as the best previous grammar-based language model. For the better of our two models these improvements are 24% and 14% respectively. We also found evidence that suggests that improvement of the underlying parser should significantly improve the model's perplexity. Since these models do not use prosodic information that most assume should help in parsing, we believe that even in the near term there is a lot of potential for improvement in immediate-head language models. Finally we note that this paper received the "Best Paper" award at ACL2001.

We now turn to our current research in the area of parsing speech data. As reported above, it is widely believed that punctuation, interjections and parentheticals all provide useful cues to phrase boundaries, and therefore their presence ought to improve parser performance. Previous experimentation with written texts had shown that removing punctuation from written texts decreases parser performance significantly, and indeed, finding prosodic cues that convey much the same information as punctuation is one of the goals of our future research. However, as a preliminary step we decided to empirically evaluate the usefulness of punctuation, interjections and parentheticals in parsing of speech transcripts. Our method of evaluation is to selectively remove each of these in turn from the training corpus, and then evaluate the accuracy of the parser's recovery of linguistically important structural details from a version of the test corpus from which the same elements were removed. Together with Donald Engel (a student at Brown),

Charniak and Johnson are systematically investigating the effect that punctuation, interjections and parethenticals have on parsing speech transcripts. As expected from the written text studies, punctuation supplies useful information for parsing spoken language transcripts, i.e., systematically removing punctuation from the training and test corpora reduces parse quality. However, contrary to the accepted wisdom, interjections and parenthetical seem not to supply useful information for parsing spoken language transcripts, i.e., systematically removing either of these elements improves parse quality, at least for our current parser. At this stage we can only speculate as to why; perhaps parentheticals are integrated into the rest of the sentence involving a structure different to the head-to-head dependency structure used in the parser, and perhaps interjections interrupt the sequences of dependencies tracked by the parser, in effect splitting the parser's internal state structure and leading to sparse data problems.

One of the central goals of this project is to integrate natural language parsing (which has been largely developed with respect to written texts) with speech recognition. As described above, we have demonstrated that parsing technology can be successfully applied to speech transcripts, and we have shown that the kinds of syntactic structures posited by a statistical parser can form the basis for a high-performance language model. These results suggest that a combined speech recognition/parsing system should perform extremely well. There is still a substantial amount of engineering and scientific work to be performed before we have achieved that integration. Currently we are investigating just what the interface between the speech recognition and parsing components should be in a combined system. It turns out that the basic data structures in each component — lattices in speech recognition, charts in parsing — are in principle quite compatible; theoretically at least one could imagine running a parser in parallel with an acoustic model (i.e., the parser would be the language model). This is a bold and attractive architecture, but we suspect that at the current stage it is impractical; the number of word hypotheses would simply overwhelm the parser. We are thus investigating ways of pruning the hypothesis space (perhaps by

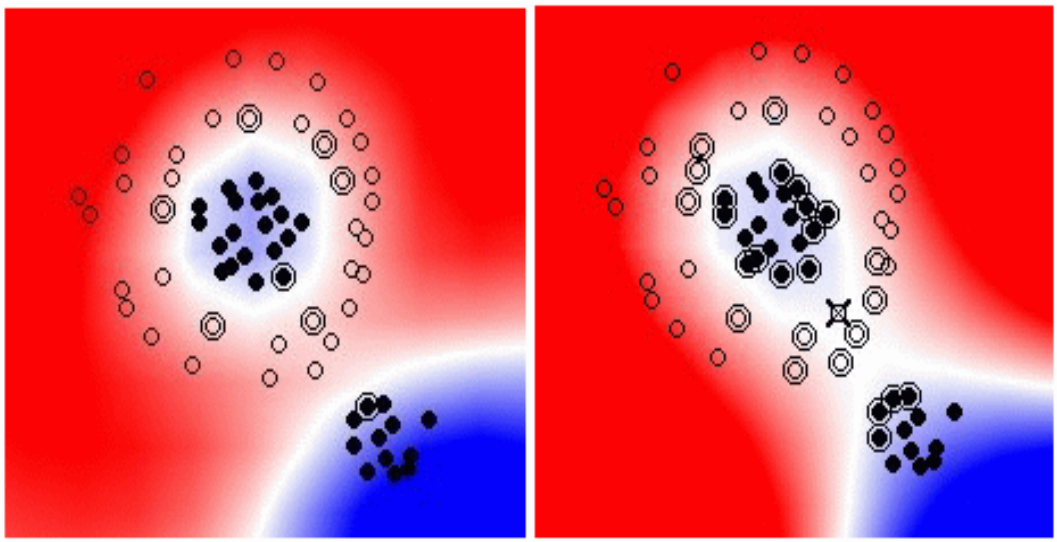

Figure 1. An SVM balances the ability to model a particular training set with generalization to other data. A feature of this machine is an ability to gracefully trade-off knowledge about the training data and the probability of error for unseen data. SVMs have proven to be very successful on several tasks including handwriting recognition, speaker identification, and vowel classification. SVMs have the ability to learn nonlinear decision regions using principles of discrimination. No assumptions about the underlying distributions are made — no parametric forms are used to build the decision surfaces.

| phone pair | SVM misclassification rate | HMM misclassification rate |
|---|---|---|
| f <=> sil | 14.6 | 13.1 |
| r <=> l | 11.9 | 17.8 |
| s <=> sil | 37.5 | 42.4 |
| s <=> z | 9.7 | 17.8 |
| t <=> p | 8.7 | 18.1 |
| t <=> d | 9.6 | 22.2 |

Table 1. A summary of performance of an SVM-based hybrid system on the most common phone confusions for Alphadigits. In some cases, the reduction in error rate is over 50%.

using a standard trigram language model) and of compacting the set of hypotheses (perhaps by using sausages instead of lattices); probably some combination of the two will turn out to be viable.

Other speech/parsing work we anticipate for this coming year will include looking at features that have been found to improve trigram language models that are not included in our language models to see if, as one might anticipate, they improve our parsing language models as well. This would include word clustering, caching, and simply training on more data (This last is not as easy for parsing models as we do not have more hand-parsed data, and thus would have to use machine-parsed data.) We also hope to start work on the integration of prosody with parsing, though this is a more ambitious project.

## F. Risk Minimization in Acoustic Modeling

An important goal in making speech recognition technology more pervasive is to improve the robustness of the acoustic models. Language models, for example, tend to port across domains much better than acoustic models. Learning paradigms for language models can fairly easily extract the domain-independent information, and don't have to deal with difficult problems such as the separation of the underlying speech spectrum from channel and ambient conditions. Though one might argue that even language models are susceptible to overtraining and a lack of generalization, the degree to which this corrupts system performance in a new domain is much less severe. Acoustic models often require extensive training or adaptation, and this, in turn,

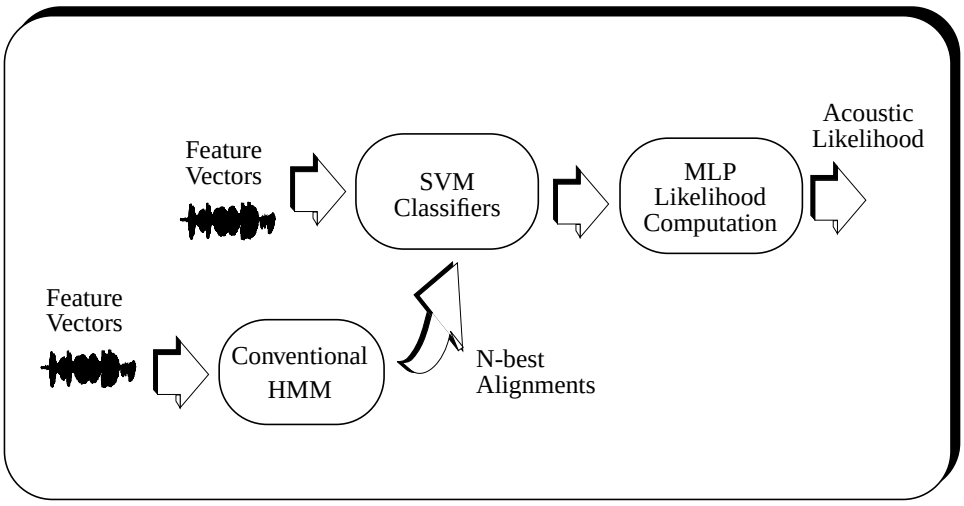


Figure 2. An overview of a hybrid HMM/SVM system being developed to improve the robustness of a speech recognition system.

requires the development of extensive application-specific data collection. The net effect is that the cost of developing new applications is very high.

A guiding principle we have in acoustic modeling is that of Occam's Razor: a model that makes less assumptions about the data will prove to be more robust. Further, we believe that we must gracefully mix representation and discrimination in our models. Intelligent machine learning seems to be a crucial issue as acoustic models can easily learn details of the acoustic channel from the training data, making them less portable to new applications where the channel, microphone, or ambient environment are different. A promising new framework for machine learning in which a balance between generalization and discrimination can be struck is based on the principle of risk minimization [12], and is known as a Support Vector Machine [13]. A summary of the benefits of the SVM approach is shown in Figure 1.

The goal in the first year of this project was to explore these models in the context of a realistic LVCSR task. Our primary focus has been kernel-based methods, which include two important related techniques: the Support Vector Machine (SVM) and the Relevance Vector Machine (RVM) [14]. On preliminary experiments involving phone classification, SVMs performed significantly better than HMMs [15]. These results are summarized in Table 1. For the six most confused phone pairs of an Alphadigit task, SVMs nearly halved the error rate, which is a significant reduction for this type of experiment.

Our initial experiments were constructed using a hybrid HMM/SVM system as shown in Figure 2. In this system, we generate N-best lists using a conventional HMM speech recognizer. We then use the same system to generate time alignments. The segments identified in these time alignments are then rescored using likelihoods generated by SVM phone classifiers. The standard Gaussian statistical models are replaced with discrimination-based SVM models.

One problem in constructing this system was how to map distances computed by the SVM classifier to posterior probabilities, which are needed by the HMM speech recognition system (more precisely, the Viterbi search engine used in the HMM-based speech recognition system). A typical solution to this problem that has been used extensively in the neural network literature is to fit a sigmoid function to the distribution of distances. However, we have recently observed that this process tends to overestimate confidence in classification. We are revisiting this issue in subsequent research described below.

We have also had to overcome a number of other mundane but important problems related to the recognition system to make these experiments possible. Because of the computational complexity of the approach, we also needed to develop an iterative training scheme in which we build classifiers on small subsets of the data and combine these classifiers (rather than training across the larger data set). We use an approach known as "chunking" [16,17] which has been shown to provide good convergence while significantly reducing computational requirements.

The SVM system overall delivered a 1% absolute (10% relative) reduction in word error rate (WER) on the Alphadigits task described above, reducing the absolute error rate from 12% to 11%. Such a small improvement is somewhat discouraging given the computational complexity of this approach. We believe a major limitation of this system is the dependence on the

HMM-based N-best lists and segmentations. Hence, we are developing approaches in which the SVM-based classifier is integrated into the training process.

A natural way to do this is to modify the concept of an SVM to incorporate probabilistic models directly. The Relevance Vector Machine (RVM) [14] attempts to overcome the deficiencies of the SVM by incorporating a probabilistic model directly into the classifier rather than using a large margin classifier [14]. The principle attraction of the RVM is that it delivers comparable performance as an SVM, but uses much fewer parameters. It is also much more computationally efficient.

A major challenge in incorporating RVM models directly into the recognition training process is the development of practical and efficient closed-loop training techniques based on EM principles that demonstrate good convergence properties. Many of these discrimination-based techniques involve some form of nonlinear optimization that is unwieldy and prone to divergence problems. We are currently developing the RVM optimization process in a Baum-Welch training framework so that the parameters of these models can be estimated in a closed-loop process on large amounts of data. We expect to complete this work in early fall of 2001.

Finally, the software being developed on this part of the project is being implemented within our public domain speech recognition system [18]. Pieces of this system will be included in our upcoming release. The core of the system consists of two new classes, SupportVectorMachine and RelevanceVectorMachine, that are part of our pattern recognition classes. We also expect to release an application note shortly describing the use of the core pattern recognition engine, and will release the hybrid system by the end of 2001. We also expect to have completed large-scale pilot experiments on spontaneous speech data at that time.

## G. REFERENCES

[1] R.D.Y. Bengio and P. Vincent, "A neural probabilistic language model," Tech. Rep. 1178, University of Montreal, 2000.

[2] M. Beckman and J. Pierrehumbert, "Intonational structure in Japanese and English," *Phonology Yearbook*, pp. 255--309, 1986.

[3] W. Byrne et al., "Pronunciation Modelling Using a Hand-Labelled Corpus for Conversational Speech Recognition," *IEEE International Conference on Acoustics, Speech and Signal Processing*, Seattle, Washington, USA, 1998.

[4] E. N'oth, 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.

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

[6] 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*, pp. 123-126, 1994.

[7] 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.

[8] 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.

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

[10] E. Charniak and M. Johnson, "Edit Detection and Parsing for Transcribed Speech," *Proceedings of the 2nd Meeting of the North American Chapter of the Association for Computational Linguistics*, Pittsburgh, Pennsylvania, USA, pp. 118-126, June 2001.

[11] E. Charniak, "Immediate-Head Parsing for Language Models," *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics*, Toulouse, France, June 2001.

[12] V. N. Vapnik, *Statistical Learning Theory*, John Wiley and Sons, New York, NY, USA, 1998.

[13] C. J. C. Burges, "A Tutorial on Support Vector Machines for Pattern Recognition," *Data*

*Mining and Knowledge Discovery*, vol. 2, no. 2, pp. 121-167, 1998.

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

[15] A. Ganapathiraju, *Support Vector Machines for Speech Recognition*, Ph.D. Dissertation, Department of Electrical and Computer Engineering, Mississippi State University, Mississippi State, Mississippi, USA, December 2001 (in preparation).

[16] E. Osuna, R. Freund, and F. Girosi, "An Improved Training Algorithm for Support Vector Machines," *Proceedings of the IEEE Workshop on Neural Networks and Signal Processing*, Amelia Island, FL, September 1997.

[17] G. Zoutendijk, *Methods in Feasible Directions — A Study in Linear and Non-linear Programming*, Elsevier Publishing Company, New York, NY, USA, 1960.

[18] M. Ordowski, N. Deshmukh, A. Ganapathiraju, J. Hamaker, and J. Picone, "A Public Domain Speech-To-Text System," *Proceedings of Eurospeech'99*, pp. 2127-2130, Budapest, Hungary, 1999.