09/01/00 — 08/31/01: 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.

A. 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 [1] 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 [1] 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 [2], 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 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.

B. 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, 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 [3], and is known as a Support Vector Machine [4]. 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) [5]. On preliminary experiments involving phone classification, SVMs performed significantly better than HMMs [6]. 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

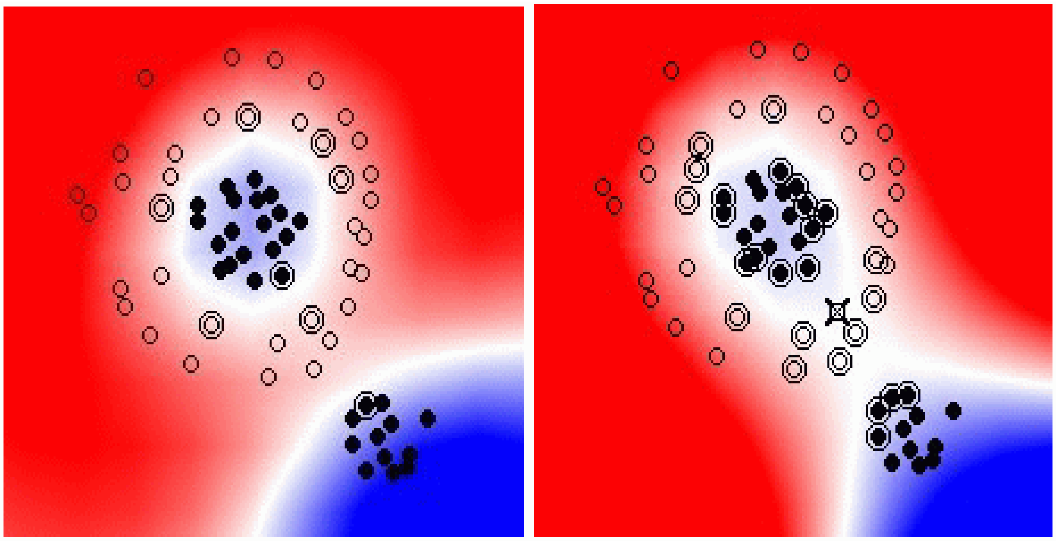

Figure 1. An SVM balances the ability to model a particular training set with generalization to other data. A feature of this machine is an ability to gracefully trade-off knowledge about the training data and the probability of error for unseen data. SVMs have proven to be very successful on several 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%.

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" [7,8] 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

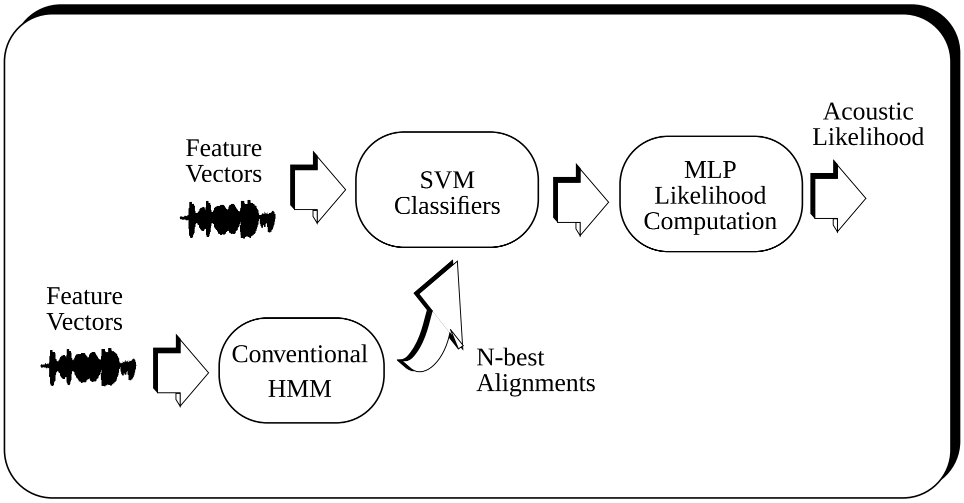

Figure 2. An overview of a hybrid HMM/SVM system being developed to improve the robustness of a speech recognition system.

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) [5] attempts to overcome the deficiencies of the SVM by incorporating a probabilistic model directly into the classifier rather than using a large margin classifier [5]. 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 [9]. 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.

C. REFERENCES

[1] 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.

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

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

[4] 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.

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

[6] 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).

[7] 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.

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

[9] 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.