Exploiting Discriminative Point Process Models for Spoken Term Detection

Atta Norouzian[1]*, Aren Jansen*[2,3]*, Richard Rose*[1]*, Samuel Thomas*[2,3]

[1]Department of Electrical and Computer Engineering, McGill University, Montreal, Canada
[2]Human Language Technology Center of Excellence,
[3]Department of Electrical and Computer Engineering, Johns Hopkins University, Baltimore, Maryland

atta.norouzian@mail.mcgill.ca,aren@jhu.edu,rose@ece.mcgill.ca,samuel@jhu.edu

Abstract

State-of-the-art spoken term detection (STD) systems are built on top of large vocabulary speech recognition engines, which generate lattices that encode candidate occurrences of each in-vocabulary query. These lattices specifiy start and stop times of hypothesized term occurrences, providing a clear opportunity to return to the acoustics to incorporate novel confidence measures for verification. In this paper, we introduce a novel exemplar distance metric to the recently proposed discriminative point process modeling (DPPM) framework and use the resulting whole word models to generate STD confidence scores. In doing so, we introduce STD to a completely distinct acoustic modeling pipeline, trading Gaussian mixture models (GMM) for multi-layer perceptrons and replacing dictionary-derived hidden Markov models (HMM) with exemplar-based point process models. We find that whole word DPPM scores both perform comparably and are complementary to lattice posterior scores produced by a state-of-the-art speech recognition engine.

Index Terms: spoken term detection, point process model, discriminative training, whole word model

1. Introduction

The goal of spoken term detection is to locate occurrences of a query term in a potentially large repository of speech recordings. The standard approach is to use a large vocabulary continuous speech recognizer (LVCSR) engine to first transform the speech into a tokenized representation and subsequently perform the detection task at the token level. Since the word error rate associated with 1-best output of these recognizers is often very high, these systems typically maintain ASR lattices containing multiple hypotheses for each utterance in order to reduce the chance of missing actual occurrences of a query term [1, 2, 3, 4]. Posterior probabilities attached to each lattice arc provide a standard confidence measure by which the putative occurrences are ranked.

LVCSR lattice posteriors provide a strong confidence measure that incorporates both acoustic and language model likelihood that a given query occurred at a particular point in the search collection. However, standard LVCSR acoustic models fail to optimally capture the durational and pronunciation variation of words in natural conversational speech. Nevertheless, a given lattice arc specifies both start and end time marks that allow us to revisit the original waveform to extract more specific information to verify the recognizer's hypothesis. There have been several recent studies that have explored this approach for STD verification [3, 4].

The point process model framework [5] for speech recognition reduces the speech signal to a sparse stream of asynchronous phonetic events and models whole syllables and words according to the temporal statistics. The discriminative point process model (DPPM) provides a means to train whole word classifiers using examples extracted from LVCSR lattices [6]. While the positive examples are naturally taken to be correct occurrences, we can limit negative examples to the incorrect hypotheses of the baseline recognizer. Much like discriminative HMM-GMM training, this strategy allows the DPPM to zero in on the recognizer confusions.

In this paper, we investigate the utility of training whole word DPPMs for rescoring word lattice arcs in the service of defining additional spoken term detection confidence measures. We build our DPPMs on top of phonetic events produced by high performance multi-layer perceptron phonetic acoustic models [7]. DPPMs rely on a suitable distance between event patterns observed for each word example. We explore the use of a novel word example distance metric, the van Rossum distance [8], borrowed from the computational neuroscience community. We find that, averaged over a broad set of query types, the van Rossum metric provides significant improvements over the binning method of [6]. The DPPM word classifiers produce scores with equal error rates approaching that of lattice posteriors generated by the state-of-the-art IBM Attila LVCSR engine [9]. Moreover, a simple linear fusion of scores produces improvement over the lattice posterior baseline. We begin with a description of the DPPM framework.

2. Whole Word Point Process Models

The point process modelling framework consists of two components: a set of phone detectors, $\{D_p\}_{p\in\mathcal{P}}$, for the set $\mathcal{P}$ of English phones, and a set of word detectors, $\{d_w\}_{w\in\mathcal{W}}$, for a lexicon $\mathcal{W}$ of words. A phone detector D_p takes as input speech signals and generates point patterns $N_p = \{t_1, t_2, ..., t_{n_p}\}$ corresponding to points in time where phone p is most clearly expressed. For a given speech segment s starting at time $t = a$ and ending at time $t = b$, the composite set of point patterns, $R_I = \{N_p\}_{p\in\mathcal{P}}$, corresponding to interval $I = (a, b]$, defines a sparse point process representation for that interval. A word detector d_w takes R_I as input and generates a score that indicates how likely it is that the interval I contains exactly w.

2.1. Phone Detectors

The process of generating point patterns for each of the phone classes is as follows. Given a speech signal of T frames, first, a feature vector time series is extracted. The feature vectors are then fed to a multilayer perceptron (MLP) phone classifier, described in Section 3.2. The MLP generates a posterior probability time series, $Y_p = y_p[1]\cdots y_p[T]$ for each phone class

$p \in \mathcal{P}$. The time series Y_p is also known as the posterior trajectory for phone p. Once the phone posterior probabilities are generated for each phone p, we apply a phone-specific filter, G_p, to the corresponding Y_p to obtain F_p. Detailed description of this process is found in [10]. The final step for deriving the point patterns, N_p, is to extract events that correspond to local maxima of F_p that are above a certain threshold, δ.

2.2. Word Models

Given the point process representation of N candidate speech intervals, $\{R_i\}_{i=1}^N$, a suitable model for word detection is a discriminatively trained classifier that provides a high score for the intervals corresponding to the actual occurrences of the word and a low score for the intervals wherein the word did not occur (false alarms). If we consider the actual occurrences of a word as positive class and all the false alarms as negative class, we have a binary-classification problem. In the following the application of a family of kernel-based classifiers to this problem is described.

2.2.1. Kernel-Based Binary Classifiers

In a typical binary-classification setting we have N data points in a d-dimensional vector space, $\{\mathbf{x}_i\}_{i=1}^N$, where each point $\mathbf{x}_i$ belongs to a class $y_i \in \{1, -1\}$. The objective is to find a function $f : \mathbb{R}^d \rightarrow \mathbb{R}$ that maps the data points into their true classes after applying an appropriate threshold. In the kernel machine framework this can be achieved by solving the optimization problem

$$f^\star = \arg \min_{f \in \mathcal{H}_K} \frac{1}{N} \sum_{i=1}^{N} V(y_i, f(\mathbf{x}_i)) + \gamma ||f||_K^2. \quad (1)$$

In Equation 1, V is a loss function and $\mathcal{H}_\mathcal{K}$ represents the reproducing kernel Hilbert space (RKHS) for the kernel function $K : \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}$. The second term on the right side of the equation is an L_2 regularizer whose purpose is to avoid overfitting with the regularization factor γ. The loss function used here is the squared loss, $V(y, f(\mathbf{x})) = [y - f(\mathbf{x})]^2$, which results in a regularized least squares (RLS) classifier. Based on the representer theorem, for a symmetric, positive semi-definite kernel K the solution to Equation 1 can be written as $f^\star = \sum_{i=1}^N \alpha_i K(\mathbf{x}_i, \mathbf{x})$. In the case of RLS the model parameters, $\alpha_i \in \mathbb{R}$, can be computed with the closed form matrix solution given by $\boldsymbol{\alpha} = (\mathbf{K} + \gamma N \mathbf{I})^{-1} \mathbf{y}$ where $\boldsymbol{\alpha} = [\alpha_1 \cdots \alpha_N]^T$, $\mathbf{y} = [y_1 \cdots y_N]^T$, $\mathbf{I}$ is the identity matrix, and $\mathbf{K}$ is the $N \times N$ Gram matrix with elements $\mathbf{K}_{i,j} = K(\mathbf{x}_i, \mathbf{x}_j)$.

We consider the class of kernel functions of the form $K(\mathbf{x}_i, \mathbf{x}_j)$=exp$(dist(\mathbf{x}_i, \mathbf{x}_j)/2\sigma^2)$, where σ is the kernel width parameter. For our problem the distance function $dist(x_i, x_j)$ is meant to represent the disimilarity between two point process patterns R_i and R_j corresponding to intervals i and j. The point process distance measures are described in the following section.

2.2.2. Distance Metrics for Point Process Patterns

The point process patterns derived from an artificial neural network based phone classifier closely resemble neural firing patterns. Therefore, we consider a measure of spike train synchrony borrowed from the computational neuroscience community and adapt it to our setting. These patterns can be regarded as continuous functions of time that take the value of one whenever an event occurs and zero otherwise. Defining a distance measure for these functions must account for two issues. First, the two event patterns will in general have different durations. Second, events occurring in two point process functions and corresponding to the same phone occurrence may not be precisely synchronized in time. There are a number of approaches for dealing with these issues [11]. Two of these approaches namely binning and van Rossum method, are investigated here.

Two sequences of events of different lengths can be aligned either by length normalization, or by dynamic programming. For both binning and van Rossum methods the alignment is performed based on length normalization. Assuming T_i corresponds to the duration of interval i with the starting time of T_0, the linear normalization maps each $t \in R_i$ to a corresponding $t' \in R_i'$ such that $t' = (t - T_0)/T_i \in [0, 1]$. This process guarantees that all the event patterns will have the same length.

Both of the distance measures considered are designed to yield a small distance between two event sequences whose event occurrence times are close but not necessarily identical. The Binning method achieves this to some extent by reducing the time resolution of the event patterns or equivalently discretizing the continuous functions. This is done by binning events into a fixed number of bins. Following this approach, each phone class p represented by a continuous function is mapped to a M-dimensional vector $\mathbf{x}_p$ whose kth component, $\mathbf{x}_p[k]$, is given by the number of events that fall in the kth bin. Afterwards, the vectors of all the phone classes of R_i (event pattern corresponding to interval i) are concatenated to form a supervector, $\mathbf{X}_i$, of dimension $M \cdot |\mathcal{P}|$, where $|\mathcal{P}|$ is the number of phone classes. The distance between two supervectors, $dist(\mathbf{X}_i \mathbf{X}_j)$, is then computed using Euclidean or cosine distance.

The binning method is limited by the trade-off between the quantization error resulting from large bins and the misalignment of events resulting from having small bins. The van Rossum method introduces an alternative approach based on smoothing the functions rather than discretizing them. Smoothing point process functions with a Gaussian filter smears each event in time. As a result of smoothing, two events from two intervals with similar but not identical occurrence times will be characterized as being close according to a simple L_2 distance. The van Rossum distance is applied to point process functions that are sampled to at least the resolution of a 10 msec analysis frame. Thus, with speech intervals assumed to be less than 2 seconds in duration, we can safely discretize time with $L = 200$ points without introducing any quantization error.

The implementation of the van Rossum method is performed in three steps. First, a Gaussian filter is applied to each phone class p of the $|\mathcal{P}|$ phone classes of the point process pattern R_i to obtain the smoothed vector, $\mathbf{b}_p$. Second, the vectors, $\{\mathbf{b}_p\}$, are concatenated to form a supervector of length $L \cdot |\mathcal{P}|$ denoted by $\mathbf{B}_i$. Figure 1 illustrates the supervectors generated using the binning and the van Rossum methods for a single speech interval containing the word "device". Third, the L_2 distance is used for computing the distance between each pair of supervectors.

3. Experimental Study

All the experiments in this paper are conducted on the Augmented Multi-party Interaction (AMI) corpus [12]. The AMI corpus consists of meeting recordings from several sites participating in the AMI project along with manually produced transcription for each of the meetings. The meeting sessions are partly naturally occurring and partly scenario-based and are in a range of domains. In this study we used a subset of the AMI cor-

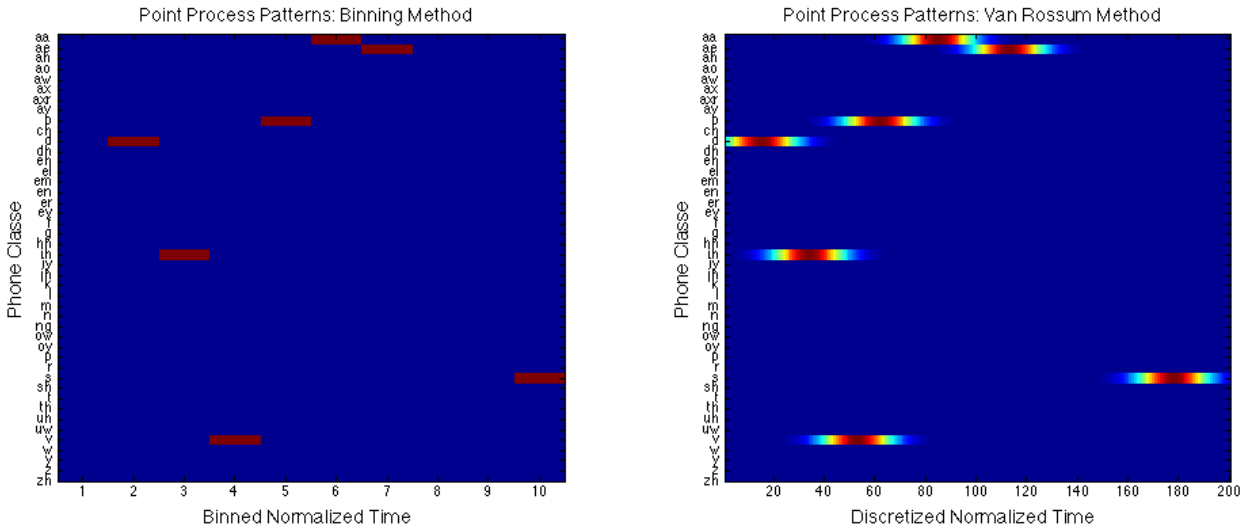

Figure 1: *Point process patterns obtained from the binning and van Rossum methods for a single speech interval containing the word "device".*

pus that was recorded using lapel microphones. This amounted to approximately 80 hours of audio recordings from 118 meetings with 121 speakers. We split the corpus into 3 equivalently sized folds, each consisting of recordings from a unique set of speakers. Folds 1 and 2 were used for training and development purposes, while fold 3 was reserved for evaluation.

We train three separate LVCSR models and three separate MLP-based phonetic acoustic models for generating lattices and point process patterns for the three folds. The first system trained on fold 1 was used for generating lattices and posteriorgrams for fold 2. The second system was trained on fold 2 and used for generating lattices and posteriorgrams for fold 1. Finally, a third system was trained on fold 1 and 2 for generating the lattices and posteriorgrams for fold 3, which are used in the evaluation. In this way none of the three systems used for generating intermediate representations suffer a test-on-train violation.

3.1. Baseline ASR system

For training and evaluating the discriminative point process models we require a set of candidate speech intervals. In this work we used word lattices generated by the state-of-the-art IBM Attila speech recognition toolkit [9], which provide candidate intervals for each word as well as baseline scores. The Attila acoustic model consists of context-dependent quinphone states built on top of 150k Gaussians. The recognizer is equipped with several state-of-the-art components: LDA and VTLN techniques were applied to the acoustic features; fMLLR and fMMI were applied for speaker adaptation; and bMMI discriminative retraining of the acoustic models was performed. Trigram language models were trained on the corpus transcripts using the appropriate folds. The acoustic model and language model were used for generating lattices with the standard Attila likelihood weighting. Using this ASR system a word error rate of 43.1% was obtained for fold 3 using a acoustic and language model trained on fold 1 and 2.

3.2. DPPM Pipeline

For evaluating the discriminative whole word DPPMs we selected a set of 35 words that had high frequencies in all three folds. The word set consists of short monosyllabic words (e.g., "*shape*") as well as long multisyllabic words (e.g., "*presentation*"). For each word in the set, we extracted all corresponding lattice arcs across all three folds. Each arc specifies a start and end time, as well as posterior probabilities.

To generate point process patterns for the candidate intervals extracted from the lattices, we must first compute phonetic posteriorgrams from which phonetic events are derived. The posteriorgrams were generated using a hierarchical configuration of multilayer perceptrons (MLPs) with two kinds of acoustic features - short-term PLP features and long-term Frequency Domain Linear Prediction (FDLP) based modulation features [7]. We use two MLPs in a hierarchical fashion for each of the two streams. The first MLP transforms acoustic features with a context of 9 frames to regular posterior probabilities. The second MLP in the hierarchy is trained on posterior outputs from the first MLP. By using a context of 23 frames, we allow the second MLP to learn temporal patterns in the posterior features. The posterior outputs of the second MLPs obtained for both acoustic features are then combined using the Dempster-Shafer rule to form the final posterior features.

The phone class posterior probabilities generated in this fashion were used to derive phonetic events using the method of [10]. Each posterior trajectory is filtered using phone specific smoothing kernels and are subsequently thresholded at $\delta = 0.24$. A peak finding algorithm is then applied to the filtered posteriors and the frames times corresponding to the peaks are extracted to form the point process patterns N_p for each phone class p.

As described in Section 2, we used RLS classifiers with RBF kernel functions for word modelling. The optimal value for the kernel width parameter σ and the regularization factor γ were tuned on the development set (fold 2), taking the values of 0.4 and $1/N$, respectively, where N is the total number of training samples for each word type. For computing the distance between two samples in the RBF kernel we presented two methods. For the binning method, we chose $M = 10$ bins for each phone class leading to 450-dimensional supervectors representing each candidate interval. For the van Rossum method, a total of 200 levels were used for quantizing the point processes

of each phone class leading to a 9000-dimension supervector, $\mathbf{B}_i$, for each candidate interval i. Each supervector $\mathbf{B}_i$ was then filtered with a Gaussian kernel with a standard deviation of 20. In Figure 2 the 9000-dimensional smooth vectors representing the 45 phone classes are shown for a number of example intervals. In this figure the horizontal red line separates the intervals corresponding to positive and negative examples.

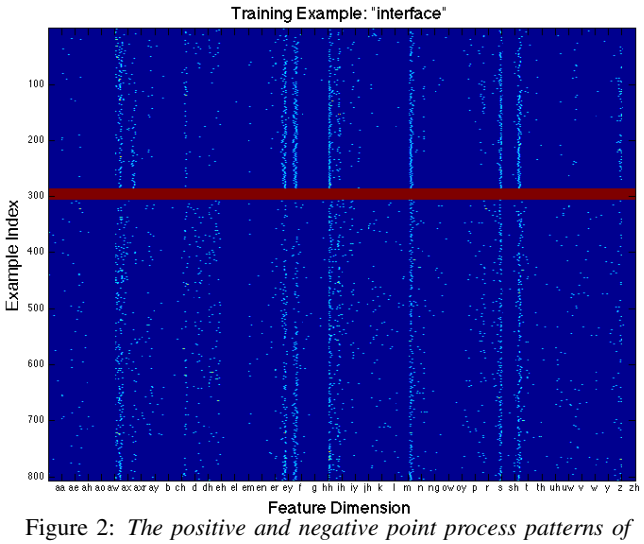

Figure 2: *The positive and negative point process patterns of candidate intervals for the word "interface".*

3.3. Performance Evaluation

We evaluated the performance of the whole word discriminative PPMs using a variety of distance functions. Furthermore, the scores obtained from DPPMs were compared to the lattice posteriors derived from acoustic model likelihoods. Finally, the fusion of the two scores was investigated. For fusion, linear regression and maximum entropy methods were examined and the linear regression method provided better results which is what is reported here. The STD results obtained from different scoring schemes, as measured by equal error rate (EER) and figure-of-merit (FOM, defined as the area under the ROC curve up to 10 false alarms/hour), are presented in Table 1. The EER and FOM values in this table are averages over the 35 words.

Several trends are apparent in the data. First, for the binning method, Euclidean distance performance was demonstrated to be equivalent to that of cosine distance. Second, the van Rossum similarity measure leads to relative improvement of 11.4% in EER over the binning method. While the van Rossum distance-derived DPPMs fall short of the lattice posterior performance, it should be noted that the DPPMs only use the knowledge derived from the acoustics in the bounds of the given lattice arc and still manage to perform admirably. Furthermore, our alternative pipeline makes no use of VTLN or speaker adaptation methods, which are known to introduce significant improvements on recognizer performance. Finally, a relative performance gain of 9.0% in EER is obtained when the lattice scores are fused with the DPPM scores, indicating a high degree of complementarity with our alternative pipeline.

4. Conclusions

We have evaluated the potential of discriminative point process models for the task of lattice rescoring in the service of spoken term detection verification. We demonstrated that DPPMs are capable of producing acoustic model scores approaching that of state-of-the-art recognizers. In addition, the DPPM scores, derived from a completely distinct pipeline, are complementary to a high performance HMM-GMM baseline. Finally, we found that the van Rossum distance between spike trains provides a more natural distance metric between phonetic event patterns, improving the performance of the original DPPM method.

Table 1: *Average EER and FOM over 35 words obtained from scores generated by various systems.*

Scoring Scheme	EER	FOM
Binning Method with Cosine Distance	14.0	81.1
Binning Method with Euclidean Distance	14.0	81.3
van Rossum Method	12.4	83.5
Lattice Posteriors	7.8	91.0
Fusion of van Rossum & Lattice Posteriors	7.1	91.6

5. Acknowledgements

The authors would like to thank Keith Kintzley of the Center for Language and Speech Processing (CLSP) for providing the phone specific filters used for smoothing the MLP-based posteriors; Damianos Karakos, also of CLSP, for his invaluable help in computing the baseline Attila lattices.

6. References

[1] A. Norouzian and R. Rose, "An efficient approach for two-stage open vocabulary spoken term detection," in *IEEE Workshop on Spoken Language Technology Proc., 194-199, 2010.* IEEE, 2010, pp. 194–199.

[2] D.R.H. Miller, M. Kleber, C. Kao, O. Kimball, T. Colthurst, S.A. Lowe, R.M. Schwartz, and H. Gish, "Rapid and accurate spoken term detection," in *Proc. Interspeech*, 2007.

[3] D. Vergyri, I. Shafran, A. Stolcke, R.R. Gadde, M. Akbacak, B. Roark, and W. Wang, "The SRI/OGI 2006 spoken term detection system," in *Proc. Interspeech*, 2007, pp. 2393–2396.

[4] Yun-Nung Chen, Chia-Ping Chen, Hung-Yi Lee, Chun-An Chan, and Lin-Shan Lee, "Improved spoken term detection with graph-based re-ranking in feature space," in *Proc. ICASSP*, 2011.

[5] A. Jansen and P. Niyogi, "Point process models for spotting keywords in continuous speech," *IEEE Trans. Audio Speech Lang. Process.*, vol. 17, pp. 1457–1470, 2009.

[6] A. Jansen, "Whole word discriminative point process models," in *Proc. ICASSP*, 2011.

[7] S. Thomas, S. Ganapathy, and H. Hermansky, "Phoneme recognition using spectral envelope and modulation frequency features," in *Proc. of ICASSP*, 2009.

[8] T. Kreuz, J.S. Haas, A. Morelli, H.D.I. Abarbanel, and A. Politi, "Measuring spike train synchrony," *Journal of Neuroscience Methods*, vol. 165, no. 1, pp. 151–161, 2007.

[9] H. Soltau et al., "The IBM 2006 GALE Arabic ASR system," in *Proc. of ICASSP*, 2007.

[10] K. Kintzley, A. Jansen, and H. Hermansky, "Event selection from phone posteriorgrams using matched filters," in *Proc. Interspeech*, 2011.

[11] C. Houghton and K. Sen, "A new multineuron spike train metric," *Neural computation*, vol. 20, no. 6, pp. 1495–1511, 2008.

[12] I. Mccowan, G. Lathoud, M. Lincoln, A. Lisowska, W. Post, D. Reidsma, and P. Wellner, "The ami meeting corpus," in *In: Proceedings Measuring Behavior 2005, 5th International Conference on Methods and Techniques in Behavioral Research. L.P.J.J. Noldus, F. Grieco, L.W.S. Loijens and P.H. Zimmerman (Eds.), Wageningen: Noldus Information Technology*, 2005.