FEATHERWEIGHT PHONETIC KEYWORD SEARCH FOR CONVERSATIONAL SPEECH

Keith Kintzley[†⋆] *Aren Jansen*[⋆] *Hynek Hermansky*[⋆]

[†] U.S. Naval Academy, Annapolis, MD, USA
[⋆] Human Language Technology Center of Excellence, Center for Language and Speech Processing
Johns Hopkins University, Baltimore, MD, USA

ABSTRACT

The point process model (PPM) for keyword search is a phonetic event-driven approach that provides a whole-word focused alternative to fast lattice matching techniques. Recent efforts in PPMs have been focused on improved model estimation techniques and efficient search algorithms, but past evaluations have been limited to searching relatively easy scripted corpora for simple unigram queries, preventing comprehensive benchmarking against standard search methods. In this paper, we present techniques for score normalization and the processing of multi-word and out of training query terms as required by the 2006 NIST Spoken Term Detection (STD) evaluation, permitting the first comprehensive benchmark of PPM search technology against state-of-the-art word and phonetic-based search systems. We demonstrate PPM to be the fastest phonetic system while posting accuracies competitive with the best phonetic alternatives. Moreover, index construction time and size are better than any keyword search system entered in the NIST evaluation.

Index Terms— point process model, spoken term detection, score normalization, compact speech indexing

1. INTRODUCTION

The point process model represents a fundamentally distinct approach to the problem of speech recognition. Given that speech arises from the highly coupled movement of articulators, a core feature of the PPM framework is the notion that words are characterized by temporal patterns of speech sounds (i.e., phonetic events). Current theories of human language acquisition also lend credence to this whole-word approach to recognition. Contrary to previous beliefs about phonemic development, a large body of evidence supports the hypothesis that infants first recognize whole words and only later construct an inventory of phonemes [1]. Additionally, the fundamental importance of temporal relations in human speech perception is corroborated by the finding that a basic neurological impairment in temporal processing lies at the root of most language learning impairment in children [2]. Beyond motivations in human speech perception, the PPM framework also possesses fundamental computational advantages. The reduction of speech to a set of distinct phonetic events produces an exceedingly sparse representation. Not only does this permit compact storage, but it also enables very fast search.

The original formulation of the point process model for keyword spotting was presented in [3]. Distinct from dense, frame-by-frame representations of speech that characterize hidden Markov model (HMM) approaches, the PPM framework operates on a sparse sequence of discrete phonetic events and words are modeled as inhomogeneous Poisson processes. This initial work presented keyword search experiments on the TIMIT dataset as well as the BU-Radio news corpus and demonstrated that the PPM system compared favorably with HMM keyword-filler approaches. A related work [4] explored an alternative method of determining phonetic events from phone posteriorgram data. It showed that the use of phonetic matched filters and appropriate threshold selection resulted in 40% fewer phonetic events and a 20% improvement word spotting performance. Capitalizing on this extremely sparse representation of speech, [5] introduced an upper bound on the PPM detection function that enabled keyword search times exceeding 500,000x faster than real-time.

Other related works have addressed the issue of estimating PPM word models. In the original presentation [3], inhomogeneous rate parameters were derived from maximum likelihood estimates (MLE) which necessitated the use of numerous keyword training examples. In [6], we demonstrated that a Bayesian approach could be applied to whole-word model estimation, significantly reducing the required number of word examples. Subsequent work presented in [7] developed improved techniques for synthesizing prior models of phonetic timing distributions using Monte Carlo and CART approaches.

Unique from previous works on this topic, here we address several challenges necessary for extending PPM techniques to the task of spoken term detection in conversational telephone speech. First, we consider approaches to modeling and search for multi-word terms as required in the 2006 NIST STD evaluation. We evaluate techniques for estimating word duration of words not present in training. Next, we address score normalization of PPM detections for subsequent evaluation under the actual term-weighted value (ATWV) metric. Finally, we present the performance of a PPM system on the 2006 NIST STD evaluation data in relation to other competitive systems.

2. PPM FOR SPOKEN TERM DETECTION

In PPM keyword search, speech is first distilled to a discrete set of points in time called phonetic events which correspond to the occurrence of phones. Typically, the acoustic signal is processed using MLP-based phone detectors that produce a phone posteriorgram representation from which phonetic events are extracted. Candidate occurrences of a keyword are identified from the PPM detection function defined as the ratio of the likelihood of a set phonetic events under a keyword model relative to its likelihood under a background model. Given a keyword w and a set of observed phonetic events $O(t)$ in the interval $(t, t+T]$, the detection function $d_w(t)$ is given by

$$d_w(t) = \log\left[\frac{P(O(t)|\theta_w, T)}{P(O(t)|\theta_{bg}, T)}\right],$$

where θ_w corresponds to the keyword-specific model parameters, θ_{bg} corresponds to background model parameters, and T is the keyword duration. This detection function is simply a log-likelihood ra-

978-1-4799-2893-4/14/$31.00 ©2014 IEEE

tio evaluated at time t which takes large values when it is likely that keyword w occurred. The PPM model hypothesizes that phonetic events within words are generated by a set of independent, inhomogeneous Poisson processes, one for each phone, that characterize the temporal structure of phones within a word. These inhomogeneous Poisson rate parameters comprise the word model θ_w. Additionally, the background arrival rate of phonetic events independent of a particular keyword is captured by a set of homogeneous Poisson processes known as the background model θ_{bg}. Keyword duration T is a latent variable, and we marginalize over T using an estimate of the keyword duration distribution.

All previous PPM keyword search studies have considered the modeling and search for single-word queries and assumed that training examples for all words were available. The 2006 NIST STD evaluation plan [8] required the search for "terms" defined as sequences of consecutively spoken words with gaps of up to 0.5 seconds allowable between words. In this section we consider the modeling of multi-word terms as single units in the PPM framework and briefly address performing multi-word queries by searching for the individual term subcomponents. Beyond the issue of multi-word search terms, the 2006 STD evaluation also necessitated the development of techniques to handle queries which do not appear in training, specifically in the PPM context, the need to estimate word duration absent any word examples. Finally, maximization of ATWV requires accurate assessment of detection confidence level; here we address the normalization of PPM detection scores.

2.1. Whole-word modeling approaches to multi-word terms

We considered four approaches to handling multi-word terms. The first and most basic is a simple concatenation of the phonetic forms of the individual terms. For example, the search term "health insurance" would be constructed from the phonetic sequence h,ɛ,l,θ,i̱,n,ʃ,ʊ,ɹ,ə,n,s. A word model is constructed directly from the phonetic sequence using equidistantly spaced Gaussian distributions with a fixed variance (see simple dictionary model in [6]). We refer to this as simple dictionary concatenation and it has the advantage of requiring no actual training examples.

In all previous work we have found that long keywords are much easier to identify than short ones, and we expect multi-word terms to be consistent with this finding. However, word model performance is also highly correlated with the number of word examples available, and it is likely that we will observe fewer examples of multi-word terms in their entirety. We have previously demonstrated in [6] that MAP estimation is an effective technique for synthesizing word models from few training examples. Therefore, beginning with a simple dictionary concatenation model prior, we then incorporate all the training examples of the term to compute a MAP-estimated whole-word model. We refer to this as a MAP-estimate model using simple dictionary prior.

Additionally, multi-word terms offer another possibility. It is very likely the case that we have many more examples of the individual words which comprise a multi-word term than we have complete examples of the multi-word term. For instance, for "health insurance" it is probable that there are numerous examples of the individual components "health" and "insurance." This offers the possibility of improving our prior model by starting with individual MAP-estimated models of the words "health" and "insurance," and then concatenating them together to form an improved prior. We refer to this as a concatenated MAP-estimated unigram prior. Finally, the few examples of the multi-word term can then be used in a new MAP-estimated model which starts from this improved prior.

To evaluate the relative performance of these approaches, we constructed an STD experiment on 230 hours of the Switchboard dataset and considered detection performance on multi-word terms. Results are listed in Table 1. While significant gains are evident between simple dictionary concatenation and the MAP-estimated model, the more sophisticated prior and subsequent MAP estimation yielded smaller improvements.

Table 1. *A comparison of multi-word modeling techniques of 571 multi-word terms on Switchboard development corpus.*

model	description	ATWV
ppm1	simple dictionary concatenation	0.4002
ppm2	MAP-estimated using simple dictionary (ppm1) prior	0.4925
ppm3	concatenated MAP-estimated unigram prior	0.5179
ppm4	MAP-estimated whole-word using unigram (ppm3) prior	0.5247

As an alternative to modeling a multi-word term in its entirety, we also considered searching for a term as the ordered union of sub-term detections with loose constraints on timing. Conceivably, this approach has two immediate advantages. First, individual words or sub-term models can be constructed independently which permits flexibility in the creation of detailed models. Second, detection of word sequences with intermediate silence is possible. Unfortunately, this method also raises a number of other issues such as how to best assign scores to multi-word detections. Additionally, conducting independent searches incurs a search speed performance hit. After some preliminary experiments, we determined that further investigation was not warranted.

2.2. Duration modeling of unseen terms

The estimation of word duration is an integral component of PPM search. In its most basic form, searching for a keyword consists of sliding a set of windows over the set of phonetic events and the evaluating the log-likelihood of events under the keyword model. Since the duration of a candidate detection is not known *a priori*, we consider a set of possible *candidate* duration windows which are drawn from an estimate of the word's duration distribution. In early PPM work with TIMIT, every keyword had 462 training examples, sufficiently many to use the empirical distribution. For later experiments on the Wall Street Journal (WSJ) corpus, the number of training examples for each keyword was much lower and use of the empirical distribution was infeasible. In its place, we adopted a parametric description of word duration based on the gamma distribution.

Handling words for which zero training examples exist requires an alternative approach, and we considered three. Admittedly crude, our first method was to compute distributions based solely on the number of phones in a word's canonical dictionary form. We simply pooled all word examples of a given phone count and computed MLE estimates for the gamma distribution parameters. For a second and more sophisticated model, we compiled duration models for all the constituent phones. Then, utilizing a technique similar to the Monte Carlo method in [7], we constructed Monte Carlo examples of word duration by sampling from the distributions of the constituent phone duration models, and then estimated MLE gamma parameters from the Monte Carlo word duration examples. Clearly, this model failed to capture any dependence of the phonetic context on phone duration.

The identical problem was addressed in [7] using a classification and regression tree (CART) approach inspired by text-to-speech synthesis work. In that work, phone duration models were estimated from the pool of examples at each node of the regression tree. Here, we opted for a simpler method to incorporate phonetic context. Our goal was to estimate phone duration models for all phonetic contexts as permitted by the number of examples available. We began by collecting pools of duration examples for each trigram phone context. Of course, many of the $O(40^3)$ possible combinations appear relatively infrequently, so if a context contained fewer than 100 examples, we backed off to the corresponding bigram phone context (and likewise from bigram to unigram). Having established the pools of examples, we then estimated MLE gamma parameters of the duration model of each context. Finally, the estimation of a word duration model proceeded as before with Monte Carlo word duration examples constructed from these context-dependent phone duration models.

To evaluate these three approaches, we considered 230 hours of Switchboard data partitioned into two folds. Assessment was based on computing the likelihood of the word durations observed in one data fold based on training data from the opposite fold using each modeling approach. Relative to the simplest duration model based on phone count, the context-dependent estimation approach provided a 27% improvement in average likelihood and was adopted for all subsequent experiments.

2.3. Score normalization

A critical element in properly assessing detections is the conversion from detection score to the estimated probability of a detection. PPM keyword detections are marked at the local maxima of the detection function (a log-likelihood ratio) as detailed in [3]. A suitable cutoff point for reliable detections varies with the number of phones in a word. In previous evaluations on TIMIT and WSJ datasets, keyword spotting performance was reported in terms of average figure of metric (FOM) and an absolute detection threshold was not required. For the 2006 STD evaluation, the performance metric is actual term-weighted value (ATWV) which requires the specification of a uniform decision threshold and a binary decision associated with each putative detection. To map PPM detection scores to a detection probability, we trained a log-linear model using keyword detections from a comparable STD experiment on Switchboard development data. In addition to PPM score, the model also used the logarithm of the keyword duration as an input parameter. These estimates of detection probability also enabled us to calculate expected counts necessary for the use of term-specific thresholding in ATWV calculation as described in [9].

3. EXPERIMENTS

Prior to testing on the 2006 STD evaluation data, we conducted extensive developmental work on a 230 hour portion of the Switchboard corpus in order to assess the methods described in the previous section (multi-word modeling, duration modeling of unseen terms and score normalization). We created a Switchboard term list with a composition roughly the same as the 2006 STD evaluation term list in percentages of multi-word terms. For acoustic models, we trained 5-layer deep neural networks to estimate posterior probabilities for 40 phonetic classes, and used them for all subsequent experiments. The 259 hours of Switchboard data was transformed into 476 dimensional FDLP-M feature vectors [10], and subsequently used to train 5 multilayer perceptrons each of size $476 \times 1500 \times 1500 \times 1500 \times 40$ using 5-fold cross validation training. We then processed the 259 hours of phone posterior data into phonetic events using phonetic matched filters as described in [4]. Finally, the data was then partitioned into two data folds for PPM training and evaluation.

Having completed developmental work on Switchboard, we then performed a series of evaluations using the NIST STD 2006 evaluation data set. The resulting XML detection list was then scored using the original NIST `STDEval` tools. STD results are reported at the bottom of Table 2 for ppm4 multi-word models (see description in Table 1) along with the results of systems in the original 2006 evaluation. In addition to STD performance, we also provide data on system processing requirements. Further, in Table 3 we provide system hardware descriptions and processor benchmark data.

3.1. Reference systems

To provide context for the PPM system performance, we have included the results from notable LVCSR and phonetic systems in the English conversational telephone speech (CTS) category of the 2006 STD evaluation (available at [11]). Overall, BBN fielded the top performing system in this category achieving an ATWV of 0.8335 [9]. The structure of BBN's system consisted of a large-vocabulary, HMM-based speech recognition system to process audio into deep word lattices upon which word posterior probabilities were estimated and a word index was generated. Multi-word term detections were determined by locating sequences of constituent words in the index that satisfied ordering and timing constraints. A key advantage of the BBN system over similar LVCSR entries came from the determination of an optimal detection threshold for each term using the expected term counts from word posterior probability estimates. Another notable entrant was the LVCSR system from IBM which achieved an ATWV of 0.7392 [12]. Both of these entries benefited tremendously from the presence of a large language model, which provided better estimates of word posterior probabilities (especially for short words) compared with systems that relied on phonetic likelihoods alone.

In contrast to the LVCSR systems, we also present two phonetic-based systems from Brno University of Technology (BUT) and Queensland University of Technology (QUT). The top performing phonetic system fielded by BUT achieved an ATWV of 0.2977. In this system the acoustic models, trained on 277 hours of primarily Switchboard data [13], were the same used in BUT's LVCSR-based primary system except that the decoding produced phoneme lattices using a phoneme bigram language model. Locating candidate detections was performed by converting the search term into a phonetic sequence using a grapheme-to-phoneme tool and then obtaining candidate sequences of overlapping phoneme trigrams from an inverted index of the phone lattice. Next, candidate sequence scores were derived from the ratio of the likelihood of the term's phone sequence to the likelihood of the best path in the phone lattice [14].

The QUT system was also based on phonetic lattice search and it yielded an ATWV of 0.0873. As described in [15], tied-state triphone HMM acoustic models were constructed using PLP acoustic features with a biphone language model to generate phonetic lattices. Next, a hierarchical index of the phone sequences and broad phone class (vowels, nasals, etc.) sequences was constructed. Query terms were converted into phonetic sequences, and then a technique termed Dynamic Match Lattice Spotting (DMLS) [16] returned putative detections of the sequences in the lattice using minimum edit distance to allow for phonetic substitutions.

In terms of performance, the PPM approach to STD falls in between that of BUT and QUT's phonetic-based entries. The QUT sys-

system	BBN	IBM	BUT	QUT	PPM t02 (GPU)	PPM a07 (non-GPU)
type	LVCSR	LVCSR	phonetic	phonetic	ppm	ppm
Indexing time (HP/HS)	16.109	7.563	86.823	18.088	**0.058**	0.610
Search speed (sec.P/HS)	**0.0014**	0.0041	13.5489	0.3300	0.0107	0.0303
Index Memory Usage (MB)	2,829.39	1,653.43	2,180.91	1,274.66	—	—
Search Memory (MB)	130.34	269.13	2.42	468.64	—	—
Index processing (HP)	48.20	22.63	259.81	54.13	**0.17**	1.83
Index size (MB)	1.17	0.98	1,528.23	1,670.52	**0.47**	0.47
ATWV	**0.8335**	0.7392	0.2977	0.0873	0.2180	0.2180

Table 2. *A comparison of NIST STD 2006 evaluation system processing resources and detection accuracy for English conversational telephone speech. For all systems, the total hours of speech (HS) is 2.99 hours.*

system	BBN	IBM	BUT	QUT	PPM t02 (GPU)	PPM a07 (non-GPU)
CPU	4 CPU Intel Xenon 3.40 GHz	4 CPU Intel Xenon 3.06 GHz	various	1 CPU Intel Pentium 4 3.00 GHz	12 CPU Intel i7-3930K 3.20 GHz	24 CPU Intel Xeon E7450 2.40 GHz
L2 Cache (KB)	1024	512	various	512	12288	12288
nbench performance						
memory index	14.497	9.946	13.967	13.459	**43.85**	27.498
integer index	9.585	9.658	11.369	9.176	**40.317**	18.921
floating-point index	18.371	16.411	19.622	18.787	**59.443**	33.399

Table 3. *NIST 2006 STD evaluation system hardware descriptions and processor benchmarks.*

tem accomplishes relatively fast lattice-based search, however, we observed that the inherently sparse representation of the PPM system permits it to search 8 times faster than DMLS with better than twice the accuracy (note: this value has been normalized based on relative processor speed benchmarks). On the other hand, the BUT approach trades speed for accuracy and achieves the best ATWV for phonetic-based systems. Yet, our PPM results are 75% of BUT's accuracy while operating 400-times faster (also normalized) with a significantly smaller footprint.

3.2. System description and processing resources

In addition to detection results, the 2006 STD evaluation also required participants to report resource and processing utilization for both indexing and search. In general, processing time is roughly 10 times slower than real-time for producing LVCSR word lattices. Phonetic lattices contain significantly more connections and require even more processing time. In the PPM system, what we call an "index" is just the collection of phonetic events. In addition to being very compact, its creation is a relatively straightforward process of feature extraction, MLP forward-pass, and matched filtering of the resulting phone posteriorgrams. The extraction of phonetic events from audio can be accomplished at roughly 17 times faster than real time. We should note that in the phonetic event production pipeline, only the MLP software currently takes advantage of the GPU; feature extraction and filtering code is not currently GPU aware. Table 2 shows both GPU and non-GPU performance.

For search, both LVCSR systems achieve very fast search times thanks to the inverted word index. Searching a phonetic lattice is a more complex endeavor [14, 16]. The BUT triphone lattice is three orders of magnitude larger than corresponding LVCSR word lattice and search is three orders of magnitude slower. The DMLS approach in the QUT phonetic system is somewhat faster. The PPM search, while fairly fast, is still basically a linear search. However, phonetic events represent an extremely sparse representation of speech, and search speed benefits because of the tiny index size. The quoted index size of 492KB for 3 hours of speech represents an uncompressed index (compression such as `gzip` provides a further 20% reduction in this case).

The extremely compact size of the PPM index is a significant advantage of our approach. It permits our system to consider extremely large volumes of audio data without being overwhelmed by either processing time or storage considerations. Additionally, the small memory footprint required by phonetic events will permit our approach to be ported to multiprocessor devices (GPU) enabling extremely fast parallel search.

In evaluating the relative system performance, it is necessary to consider the computation speed of the systems at the time of the original evaluation. To offer some perspective on the relative speed, we present system descriptions and benchmarks in Table 3. Overall the `t02` GPU machine is roughly 3-4 times faster than 2006-era machines and `a07` is approximately twice as fast.

4. CONCLUSIONS

In this work we have addressed many of the technical challenges required to enable the PPM system to accomplish spoken term detection. Furthermore, this study provides the first side-by-side comparison of a PPM system for spoken term detection in the context of other well documented systems on a standard evaluation dataset. Unquestionably, LVCSR-based systems will outperform systems that do not currently benefit from a language model. Yet, we clearly observe that PPM keyword spotting achieves performance results competitive with other state-of-the-art phonetic-based systems. More significantly, PPM keyword spotting accomplishes this while requiring a fraction of the computational and storage resources

5. REFERENCES

[1] Johannes C Ziegler and Usha Goswami, "Reading acquisition, developmental dyslexia, and skilled reading across languages: a psycholinguistic grain size theory.," *Psychological bulletin*, vol. 131, no. 1, pp. 3, 2005.

[2] Paula Tallal, Steve Miller, and Roslyn Holly Fitch, "Neurobiological basis of speech: a case for the preeminence of temporal processing," *Annals of the New York Academy of Sciences*, vol. 682, no. 1, pp. 27–47, 1993.

[3] A. Jansen and P. Niyogi, "Point process models for spotting keywords in continuous speech," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 17, no. 8, pp. 1457 – 1470, nov. 2009.

[4] K. Kintzley, A. Jansen, and H. Hermansky, "Event selection from phone posteriorgrams using matched filters," in *Proceedings of INTERSPEECH*. ISCA, 2011, pp. 1905–1908.

[5] K. Kintzley, A. Jansen, K. Church, and H. Hermansky, "Inverting the point process model for fast phonetic keyword search," in *Proceedings of INTERSPEECH*. ISCA, 2012, pp. 2438–2441.

[6] K. Kintzley, A. Jansen, and H. Hermansky, "MAP estimation of whole-word acoustic models with dictionary priors," in *Proceedings of INTERSPEECH*. ISCA, 2012, pp. 787–790.

[7] K. Kintzley, A. Jansen, and H. Hermansky, "Text-to-speech inspired duration modeling for improved whole-word acoustic models," in *Proceedings of INTERSPEECH*. ISCA, 2013.

[8] National Institute of Standards and Technology, "The Spoken Term Detection (STD) 2006 Evaluation Plan," September 2006, http://www.itl.nist.gov/iad/mig//tests/std/2006/docs/std06-evalplan-v10.pdf.

[9] David RH Miller, Michael Kleber, Chia-Lin Kao, Owen Kimball, Thomas Colthurst, Stephen A Lowe, Richard M Schwartz, and Herbert Gish, "Rapid and accurate spoken term detection," in *Proceedings of INTERSPEECH*, 2007, pp. 314–317.

[10] S. Ganapathy, S. Thomas, and H. Hermansky, "Comparison of modulation features for phoneme recognition," in *Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on*, 2010, pp. 5038–5041.

[11] National Institute of Standards and Technology, "Results of the Spoken Term Detection (STD) 2006 Evaluation," December 2006, http://www.itl.nist.gov/iad/mig/tests/std/2006/pubdata/std06_results_20061207.tgz.

[12] Jonathan Mamou, Bhuvana Ramabhadran, and Olivier Siohan, "Vocabulary independent spoken term detection," in *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 2007, pp. 615–622.

[13] Thomas Hain, Lukas Burget, John Dines, Giulia Garau, Martin Karafiat, Mike Lincoln, Jithendra Vepa, and Vincent Wan, "The AMI meeting transcription system: Progress and performance," in *Machine Learning for Multimodal Interaction*, Steve Renals, Samy Bengio, and Jonathan Fiscus, Eds., vol. 4299 of *Lecture Notes in Computer Science*, pp. 419–431. Springer Berlin Heidelberg, 2006.

[14] Lucas Burget, Honza Cernocky, Michal Fapso, Martin Karafiat, Pavel Matejka, Petr Schwarz, and Igor Szoke, "Indexing and search methods for spoken documents," in *Text, Speech and Dialogue*, Petr Sojka, Ivan Kopecek, and Karel Pala, Eds., vol. 4188 of *Lecture Notes in Computer Science*, pp. 351–358. Springer Berlin Heidelberg, 2006.

[15] Roy G Wallace, Robert J Vogt, and Sridha Sridharan, "A phonetic search approach to the 2006 NIST Spoken Term Detection Evaluation," in *Proceedings of INTERSPEECH*. ISCA, 2007, pp. 2385–2388.

[16] K. Thambiratnam and S. Sridharan, "Dynamic match phone-lattice searches for very fast and accurate unrestricted vocabulary keyword spotting," in *Acoustics, Speech, and Signal Processing, 2005. Proceedings. (ICASSP '05). IEEE International Conference on*, 2005, vol. 1, pp. 465–468.