Rapid and Accurate Spoken Term Detection

David R. H. Miller, Michael Kleber, Chia-lin Kao, Owen Kimball
Thomas Colthurst, Stephen A. Lowe, Richard M. Schwartz, Herbert Gish

BBN Technologies, Cambridge MA 02138, USA
{drmiller,mkleber,ckao,okimball,tcolthur,slowe,schwartz,gish}@bbn.com

10.21437/Interspeech.2007-174

Abstract

We present a state-of-the-art system for performing spoken term detection on continuous telephone speech in multiple languages. The system compiles a search index from deep word lattices generated by a large-vocabulary HMM speech recognizer. It estimates word posteriors from the lattices and uses them to compute a detection threshold that minimizes the expected value of a user-specified cost function. The system accommodates search terms outside the vocabulary of the speech-to-text engine by using approximate string matching on induced phonetic transcripts. Its search index occupies less than 1Mb per hour of processed speech and it supports sub-second search times for a corpus of hundreds of hours of audio. This system had the highest reported accuracy on the telephone speech portion of the 2006 NIST Spoken Term Detection evaluation, achieving 83% of the maximum possible accuracy score in English.

Index Terms: spoken term detection, keyword spotting, word spotting, audio indexing

1. Introduction

Finding instances of a particular spoken word or phrase in a corpus of audio recordings is one of the fundamental problems of automated speech processing. The task has a history stretching back more than 35 years and has gone under many names, including "word-spotting," "audio indexing," and "spoken term detection." Early approaches focused on building custom detectors, either template-based or probabilistic, for prespecified words [1]. For the past fifteen years, approaches that couple speech-to-text (STT) technology with traditional text-matching techniques have been more successful for both predefined and *ad hoc* search of a complete corpus [2] [3].

In this paper, we present a multi-lingual spoken term detection system for conversational telephone speech that BBN Technologies constructed in response to the NIST Spoken Term Detection (STD) evaluation of 2006 [4]. The system follows the transcribe-and-text-match paradigm, but uses word lattices instead of single transcripts as the STT engine output in order to mitigate transcription errors. It estimates the posterior probability of each detection's correctness directly from the lattices and uses it to balance false positive and false negative errors in accordance with user-defined costs. To handle out-of-vocabulary searches, the system performs an approximate search of a phonetic transcript.

Our approach to in-vocabulary searches is most similar to that followed in [5]. However, we use uncollapsed word lattices to compute word posteriors instead of word confusion networks. Both approaches are based on the log-likelihood ratio scoring method [6], originally presented for N-best hypothesis lists. Our approach to indexing the full lattices is akin to [7], but simplified because we are doing term detection, not spoken document retrieval.

The system performed well in all three languages in the 2006 NIST STD Evaluation. For each language, it had the highest reported accuracy on the telephone speech portion of the evaluation corpus, achieving 83% of the maximum possible score for English. The system also exhibited good operational characteristics: it executed extremely fast searches based on a small index, and consumed a moderate amount of computation for speech-to-text and indexing.

2. Task description

The work presented here addresses the Spoken Term Detection task defined by NIST for the 2006 STD Evaluation [4].

In the NIST STD task, a system takes a corpus of recorded speech files and creates an index. It then accepts textual search terms and uses the index to produce sets of ⟨file, time⟩ pairs which it asserts correspond to spoken instances of each term. Accuracy is judged relative to a time-marked reference transcript. A system assertion is considered correct if a corresponding exact orthographic match of the term appears in the reference transcript within 0.5 seconds of the asserted time.

Terms are presented in the native orthography of the language (English, Arabic, or Mandarin) and are neither known at indexing time nor constrained to come from any particular closed vocabulary. Terms may be single words or arbitrary multi-word strings, in which case assertions must exactly match an uninterrupted word sequence in the reference transcript to be considered correct.

System accuracy on a given collection of query terms is measured by a new metric, constructed to reflect one potential application of an STD system. "Actual Term-Weighted Value" (ATWV) is defined in [4] as

$$\text{ATWV} = \text{mean}\left(\frac{N_{\text{correct}}(s)}{N_{\text{true}}(s)} - \beta \cdot \frac{N_{\text{spurious}}(s)}{T - N_{\text{true}}(s)} \right), \quad (1)$$

where the search term s occurs $N_{\text{true}}(s)$ times in the reference transcript and the system makes $N_{\text{correct}}(s)$ correct and $N_{\text{spurious}}(s)$ incorrect assertions of s. T is the total duration of the audio corpus in seconds. The parameter β incorporates the relative costs of misses and false assertions and the prior probabilities of search terms; it was set to 999.9 for the evaluation. To avoid division by zero, the mean is taken over only the terms in the set for which $N_{\text{true}}(s)$ is positive.

3. System description

The spoken term detection system we constructed has four components: a speech-to-text engine, an indexer, a detector, and a

decider. The speech-to-text engine processes audio files and outputs word lattices and single-best phonetic transcripts. The indexer takes these as input and creates an index containing a precomputed list of candidate detection records for each word in the speech-to-text lexicon. The index also contains the phonetic transcripts to accommodate out-of-vocabulary search terms. The detector loads the index and processes a list of search terms, generating a sorted, scored list of detection records for each term. Finally, the decider takes the lists of candidate detections and the cost parameter β and sets a per-term score threshold for making yes/no decisions.

We used the same overall system design for English, Levantine, and Mandarin, and customized for each condition only the STT engine, the text-to-phoneme engine for OOV terms, and some tuned system parameters.

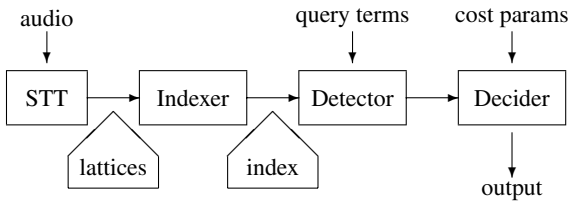

Figure 1: *BBN Spoken Term Detection System architecture.*

3.1. Recognition

The first stage of the system uses a traditional speech-to-text engine. It automatically segments the audio corpus into utterances and produces for each one a word lattice and the phoneme transcript corresponding to the single best path through that lattice. The lattice arcs are annotated with acoustic and language model probabilities from the final pass of adapted decoding.

We tried two different English configurations of the BBN Byblos STT recognizer. Our baseline was the configuration described in [8]. We also experimented with the configuration described in [9], which runs approximately 10 times faster but has a higher word error rate. In each case, we trained the acoustic models on the 2300-hour EARS RT04 CTS training corpus [10] and the language models on approximately 1 billion words of data available from the Linguistic Data Consortium (LDC) and the University of Washington (UW)[11].

For the Mandarin and Levantine systems, we used the configuration of Byblos described in [12], but simplified to omit system combination. We trained the Mandarin system using roughly 260 hours of CTS audio and 240 million words of textual data available from LDC and UW. We trained the Levantine acoustic models using 57 hours of speech compiled by LDC, and trained the language model from the transcripts of 250 hours of data. We did not have a large phonetic dictionary for Arabic, so we relied on a "grapheme-as-phoneme" approach, in which words get a pronunciation equal to their spelling, plus hand-crafted phonetic spellings for 100 high frequency words [12].

3.2. Indexing

Following recognition, our system processes the collection of word lattices and precomputes a set of detection candidates for each word $w_1, w_2, \ldots, w_L$ in the STT lexicon. For each instance of w_i in a given lattice, the system estimates the posterior probability of correctness to be the fraction of the total lattice likelihood that flows through the edge corresponding to that instance of w_i. It then clusters instances of w_i that occupy approximately the same time interval and sums their posteriors to get an overall posterior for a single representative detection candidate for w_i. The detection candidates from all lattices are accumulated into L independent lists. The indexing module sorts each list by posterior and constructs a persistent index comprising the lists and a hash map from word w_i to the corresponding list. The lattices themselves are discarded.

No indexing is performed on the STT phonetic transcripts, which are passed on to the retrieval module unchanged.

3.3. Detection

After recognition and indexing are complete, the system is ready to process *ad hoc* search terms. The detection module produces a list of scored candidates in response to a search. For single-word in-vocabulary terms, it simply retrieves the precomputed list of candidate detections from the index. For query terms consisting of multiple in-vocabulary words, it retrieves each word list individually, then finds strings of single-word detections that occur in the correct order and without long temporal gaps. To any such string, the system assigns a proxy posterior probability equal to the minimum posterior of the component word detections.

This handling of multi-word terms is approximate in two ways. Since we have discarded the original lattices, it is possible to reconstruct a string which never actually occurred as a complete hypothesis in the lattice. And for those cases where the string did occur in the lattice, the fraction of lattice likelihood flowing through the corresponding path may be less than the minimum fraction flowing through its constituent words. In practice, we found that these flaws did not hurt accuracy (see Section 4).

For query terms including even a single out-of-vocabulary word, the system follows a different detection logic. It hypothesizes a pronunciation of each OOV query term, then invokes the TRE agrep package [13] to search for local alignments between the pronounced query and the 1-best phonetic transcripts that have small edit distance. For English, we used the t2p text-to-phoneme package [14]; for Mandarin, we used a dictionary-based character-to-phoneme scheme; for Levantine, we used a direct grapheme-to-phoneme mapping.

3.4. Decision

The STD task defined in Section 2 scores systems based on a binary decision of which candidate detections to assert and which not. In general, if an evaluation metric assigns marginal benefit B to each correct assertion and marginal cost C to each incorrect assertion, then a system should assert only those candidates whose probability of correctness p implies that the expected value $pB - (1-p)C$ is positive. For the ATWV metric (1), $B = 1/N_{\text{true}}$ and $C = \beta/(T - N_{\text{true}})$, giving the threshold

$$p > \frac{N_{\text{true}}}{T/\beta + \frac{\beta-1}{\beta} N_{\text{true}}}, \tag{2}$$

where $\frac{\beta-1}{\beta} \approx 1$, and $T/\beta \approx 10$ for three hours of audio. Since $N_{\text{true}}(s)$ is unknown, the system estimates it as the sum of the posterior estimates for all candidate detections of s anywhere in the corpus, scaled by a term-independent learned factor to account for occurrences that were pruned before lattice generation.

For out-of-vocabulary terms, our system offers no reasonable proxy for posterior probability, so we abandon (2). Instead, it asserts the best k candidates whose phonetic edit distance is less than some fraction f of the query pronunciation length, where f and k are term-independent thresholds we optimized on development data.

4. Experiments

We performed development experiments and analysis using English and Levantine Corpora provided by NIST and a Mandarin corpus prepared by BBN. We also participated in the NIST STD 2006 Evaluation, and report NIST's scoring of our system's performance on the evaluation data (see Table 1). We worked only on the conversational telephone speech (CTS) portion of each data set. NIST supported two versions of the Levantine task, one including diacritical markings and one ignoring them; we worked only on the non-diacritized task.

4.1. Audio corpora and query sets

The English CTS development corpus provided by NIST comprised three hours of conversations from the Fisher English collection; the foreign language development corpora comprised only one hour of data from the Fisher Arabic and Mandarin Callfriend collections. The evaluation corpora were of the same length as the development sets and mostly drawn from the same sources (in Mandarin, the NIST evaluation set was drawn from HKUST; for consistency, BBN worked with an HKUST-based development corpus constructed in-house).

For each audio corpus, NIST also provided a list of approximately 1000 query terms, constructed using the audio's reference transcript as a guide but also including some out-of-corpus queries. The majority of queries were single words; about 10% were 3- or 4-word phrases. English query terms from the development set included 'yeah,' 'point,' 'Baghdad,' 'organizing committee,' and 'relief pitcher Mariano Rivera.'

4.2. Results

Table 1 shows the accuracy, size, and speed of our system for both the development and evaluation data sets. These were the highest CTS evaluation results reported by NIST in each language; our two English systems were the two highest scorers.

Lattices Searching STT lattice output instead of 1-best word transcripts offers greater recall, but risks generating more false alarms. Table 2 compares our baseline system with one that indexes 1-best transcripts with posteriors derived from an N-best list via a general linear model [15]. While the number of candidate words in the index rises dramatically for lattices, false positives are kept in check by accurate word posteriors.

	Development		NIST STD Eval		
System	ATWV	WER	ATWV	speed	size
English	0.852	14.9%	0.833	43.0	1.0
Eng faster	0.766	18.1%	0.761	2.7	0.5
Mandarin	0.343	31.7%	0.381	15.0	0.8
Levantine	0.410	43.3%	0.347	9.5	1.4

Table 1: *Results on development data and from NIST 2006 STD Evaluation. Speed of recognition and indexing in CPU hours per audio hour; size of index in Mb per audio hour. Average search times were under 0.01s per term.*

	English	Eng faster	Mandarin	Levantine
ATWV:				
1-best	0.754	0.711	0.228	0.242
lattice	0.852	0.766	0.343	0.410
words per hour:				
1-best	10,713	10,607	8,667	7,089
lattice	86,422	30,713	67,503	153,689

Table 2: *Gains from searching lattices versus 1-best transcripts. ATWV improves markedly because words outside the 1-best may still be likely enough to warrant asserting them.*

Multi-word terms In Section 3.3 we described an approximate method for detecting multi-word query terms and computing their posterior probabilities. Table 3 compares this approximate method to the exact search, which asserts only a complete occurrence of a multi-word term in an STT lattice and assigns it a posterior probability equal to the fraction of the lattice likelihood that flows through that path.

The approximate method uses a much smaller index than the exact method. We were initially surprised to find that ATWV actually increased slightly when using the "weakest-link" posterior approximation. However, analysis showed that approximate matching sometimes recovered from decoding errors in which all words of a phrase were recognized individually, but did not appear as an unbroken hypothesis in the lattice.

	English		Mandarin		Levantine	
Search	ATWV	size	ATWV	size	ATWV	size
exact	0.829	33	0.323	11	0.363	51
approx	0.839	0.9	0.335	0.8	0.376	1.5

Table 3: *Approximate multi-word phrase searches (Section 3.3) allow a smaller index with no apparent loss of accuracy. ATWV on development data; size of index in Mb per hour of audio.*

Pipeline attrition Flaws in STT lattices, in ranking of detections, and in thresholding all contribute to a final ATWV less than 1.0. We used two additional metrics to diagnose where our system lost accuracy.

For each query term s, the decision procedure takes a list of potential detections ordered by estimated posterior probability and picks some cutoff threshold $\theta(s)$ in an attempt to maximize ATWV. There is, of course, some optimal threshold $\theta_O(s)$ which truly does maximize ATWV for the given ordered list of candidate detections. We refer to the score associated with $\theta_O(s)$ as the Oracular Term-Weighted Value, or OTWV.

Even the decision induced by $\theta_O(s)$ may include misses or false alarms, when some incorrect detection has higher estimated posterior than some correct detection. If we ignore the penalty for false alarms, we can see the full value of all true hits present in the lattices and phonetic transcripts. We call the result the no-FA score.

System	no-FA	OTWV	ATWV
English	0.957	0.902	0.852
Eng faster	0.874	0.811	0.766
Mandarin	0.554	0.440	0.343
Levantine	0.672	0.524	0.410

Table 4: *Value with perfect ordering and thresholding (no-FA), with oracle thresholding of our ordering (OTWV), and with our actual ordering and thresholding (ATWV).*

Table 4 shows our system's accuracy on development data as measured by all three scores. Reading across the rows, one can see the remaining available value after the speech recognition, detection ranking, and thresholding stages.

IV versus OOV accuracy Our IV processing is very different from our OOV processing. Detection accuracy for out-of-vocabulary terms was uneven across languages and between development and evaluation data. The system was most successful on Levantine (see Table 5): the OOV ATWV was one-third of the IV ATWV on development data, while on the evaluation data the OOV ATWV was *better* than the IV ATWV. By contrast, in Mandarin we were unable to find a global threshold on phonetic edit distance that produced a positive ATWV for OOV terms, and so the system asserted nothing for these queries. For English, there are too few OOV terms in the query sets to gain any insight.

Levantine Data Set	IV	OOV	all
Development	0.441	0.162	0.410
Eval	0.345	0.364	0.347

Table 5: *ATWV for in-vocabulary vs. out-of-vocabulary terms.*

5. Discussion

Much of the power of our system comes from asserting hypotheses that appear only in the STT lattices and not in the 1-best transcript (Table 2). To avoid being swamped by false alarms, the system needs an accurate confidence estimate on these sub-maximal hypotheses. In order to achieve good results, we found that we needed to tune the relative weights of the acoustic and language model likelihoods as usual, and also tune an overall scaling factor applied to the combined log-likelihood. This overall scaling does not alter the rankings of competing hypotheses for an utterance, but changes how smoothly confidence is distributed across hypotheses.

The "decider" component of our system is the only one specific to the ATWV metric defined by NIST. We found it critical to have a term-specific decision threshold rather than a global cutoff for all terms. Because the benefit of correctly finding a term is inversely proportional to the frequency of that term, the ATWV metric heavily emphasizes recall of rare terms. This emphasis is so pronounced that our system will assert the best single candidate even when its posterior is extremely low. The emphasis on rare terms also affects how ATWV scales with corpus size: uniquely-occurring terms often remain unique as the corpus grows, and at the same time more uniquely-occurring terms are introduced. The result is a higher ATWV for a large corpus than for the same audio and query terms split into several smaller corpora. None of these characteristics of ATWV cause problems for our system, as the decider logic handles the maximization correctly.

The development and evaluation audio corpora were quite small: 3 hours for English, 1 hour each for Mandarin and Levantine (though there were 1000 queries for each language). While we believe that the accuracy measured on these sets is predictive of accuracy on large corpora, we are less sanguine about extrapolating speed measurements to thousands of hours. Using standard UNIX tools we benchmarked our retrieval for an in-vocabulary search at less than 0.01 seconds per term for a 3 hour corpus, with much of that time spent on overhead processing that will not scale with the size of the corpus. At this point, all we can confidently project is that the system can accommodate many hundreds of hours of audio and still deliver sub-second in-vocabulary search. Our out-of-vocabulary search operated linearly on the data. Application of standard indexing techniques should speed it considerably.

The overall ATWV results for English are far superior to those in Mandarin or Levantine. While much of the gap reflects the difference in underlying STT word error rate, some is caused by problems in foreign language transcription. The evaluation criteria for correct detection included exact orthographic match (including word breaks) between a target term and the reference transcript. However, the Levantine data contains numerous cases where the same spoken word is transcribed with different orthographies (even ignoring diacritics), and the Mandarin data contains widespread errors and inconsistencies in word segmentation. To a smaller extent, these problems are also present in the English transcription, primarily in hyphenation and abbreviation conventions.

Acknowledgments The authors would like to thank John Makhoul for his help with Levantine transcription issues.

6. References

[1] Rohlicek, J.R., "Word Spotting" in *Modern Methods of Speech Processing*, Ramachandran, R. and Mammone, R. (Eds.), Kluwer International Series in Engineering and Computer Science, Kluwer Academic Publishers, Boston, 1995.

[2] Weintraub, M., "Keyword-Spotting Using SRI's DECIPHER Large-Vocabulary Speech-Recognition System", in *Proc. IEEE ICASSP*, 1993.

[3] Makhoul, J. et al., "Speech and Language Technologies for Audio Indexing and Retrieval", *Proc. of the IEEE*, Vol. 88, No. 8, August 2000.

[4] http://www.nist.gov/speech/tests/std/

[5] Mamou, J., Carmel, D., Hoory, R., "Spoken Document Retrieval from Call-Center Conversations", in *Proc. SIGIR*, 2006.

[6] Weintraub, M., "LVCSR Log-likelihood Ratio Scoring for Keyword Spotting", in *Proc. IEEE ICASSP*, 1995.

[7] Chelba, C. and Acero, A., "Position Specific Posterior Lattices for Indexing Speech", in *Proc. ACL*, 2005.

[8] Zhang, B., Matsoukas, S., Schwartz, R., "Discriminatively trained region dependent feature transforms for speech recognition", in *Proc. IEEE ICASSP*, 2006.

[9] Matsoukas, S., et al., "The 2004 BBN 1xRT Recognition Systems for English Broadcast News and Conversational Telephone Speech", in *Proc. Interspeech*, 2005.

[10] http://projects.ldc.upenn.edu/EARS

[11] http://ssli.ee.washington.edu/projects/ears/WebData

[12] Abdou, S., et al., "The 2004 BBN Levantine Arabic and Mandarin CTS transcription systems", DARPA EARS workshop, Palisades, NY, Nov, 2004.

[13] Laurikari, V., http://laurikari.net/tre

[14] Lenzo, K., http://www.cs.cmu.edu/∼lenzo/t2p

[15] Siu, M. and Gish, H., "Improved estimation, evaluation, and applications of confidence measures for speech recognition", in *Proc. EuroSpeech*, 1997.