nexidia

WHITE PAPER

Phonetic Search Technology

A Whitepaper by Nexidia, Inc.

Copyright Notice

Nexidia Inc. – Headquarters
3565 Piedmont Road NE
Building Two, Suite 400
Atlanta, GA 30305
USA

404.495.7220 tel
404.495.7221 fax
866.355.1241 toll-free

Nexidia Ltd. – UK
Gainsborough House
2 Sheen Road
Richmond TW9 1AE
United Kingdom

+44 (020) 8973 2440 tel
+44 (020) 8973 2301 fax

nexidia.com

Introduction

From call centers to broadcast news programs, the quantity of digital files being created is growing quickly and shows no signs of slowing. While valuable information may exist in the audio of these files, there has historically been no effective means to organize, search and analyze the data in an efficient manner.

Consequently, much of this data was unavailable for analysis, such as the millions of hours of call center calls recorded every year that are archived for legal reasons. Using a more traditional approach, a very small amount of audio may be listened to, but in an ad-hoc manner, such as random audits by call center managers, or listening to various broadcasts. Targeted searching, however, is difficult. If this audio data were easily searchable, many applications would be possible, such as: reviewing only calls that meet set criteria, performing trend analysis across thousands of hours of customer calls, searching an entire newscast to find the exact location where a certain topic is discussed and many other uses.

The difficulty in accessing the information in most audio today is that unlike some broadcast media, closed captioning is not available. Further, man-made transcripts are expensive to generate, and limited in their description. Audio search based on speech-to-text technology is not scalable, depends on highly trained dictionaries and generates a prohibitive total cost of ownership. What is needed is an alternate approach.

In this paper, we summarize prior work in searching audio and examine characteristics of the various methods. We then introduce and describe a better approach known as phonetic-based search, developed by the research group at Nexidia in conjunction with the Georgia Institute of Technology. Phonetic-based search is designed for extremely fast searching through vast amounts of media, allowing search for words, phrases, jargon, slang and other words not readily found in a speech-to-text dictionary. Below we provide a description of Nexidia's technology and then we discuss the accuracy of phonetic search and finally we present current applications of the technology itself.

Contact centers/enterprise, rich media, legal/audio discovery and government applications are areas where Nexidia has been successfully applied.

Prior Work in Audio Search

Retrieval of information from audio and speech has been a goal of many researchers over the past ten years. The simplest solution to this problem would be to use Large Vocabulary Continuous Speech Recognition (LVCSR), perform time alignment, and produce an index of text content along with time stamps. LVCSR is sufficiently mature that toolboxes are publicly available such as HTK (from Cambridge University, England), ISIP (Mississippi State University, USA), and Sphinx (Carnegie Mellon University, USA) as well as a host of commercial offerings. Much of the improved performance demonstrated in current LVCSR systems comes from better linguistic modeling [Jurafsky] to eliminate sequences of words that are not allowed within the language. Unfortunately, the word error rates are seldom zero.

The need for better automatic retrieval of audio data has prompted formulation of databases specifically to test this capability [Graff]. Also, a separate track has been established for spoken document retrieval within the annual TREC (Text Retrieval Conference) event [Garofolo]. An example can be seen in [Johnson]. In this research, a transcription from LVCSR was produced on the NIST Hub-4 Broadcast News corpus. Whole sentence queries are posed, and the transcription is searched using intelligent text-based information extraction methods. Some interesting data from this report shows that word error rates range from 64% to 20%, depending on the LVCSR system used, and closed captioning error rates are roughly 12%. While speech recognition has improved since these results, the improvement has been incremental.

Ng and Zue [Ng] recognized the need for phonetic searching by using subword units for information retrieval. Although the phonetic error rates were high (37%) and performance of the retrieval task was low compared to LVCSR methods, great promise was anticipated by the authors.

In the LVCSR approach, the recognizer tries to transcribe all input speech as a chain of words in its vocabulary. Keyword spotting is a different technique for searching audio for specific words and phrases. In this approach, the recognizer is only concerned with occurrences of one keyword or phrase. Since the score of the single word must be computed (instead of the entire vocabulary), much less computation is required. This was very important for early real-time applications such as surveillance and automation of operator-assisted calls [Wilpon] [Wohlford].

Another advantage of keyword spotting is the potential for an open vocabulary at search time, making this technique useful in archive retrieval. This technique, however, is inadequate for real-time execution. When searching through tens or hundreds of

INDUSTRY	BENEFITS FROM PHONETIC SEARCH
Contact Centers/Enterprise	> Improved customer interactions > Deeper business intelligence > Operational efficiencies
Rich Media	> Large amounts of long form content is searchable > Automated categorization and filtering > Synchronize stories with videos > Ad targeting > Easily monetized content
Legal/Audio Discovery	> Corporate compliance > Litigation support > Fast and accurate audio discovery
Government	> Audio search > Public safety > Standards compliance

thousands of hours of archived audio data, scanning must be executed many thousands of times faster than real-time. To achieve this goal, a new class of keyword spotters has been developed that performs separate indexing and searching stages. In doing so, search speeds that are several thousand times faster than real time have been successfully achieved.

The two predominant approaches have been fast-indexing methods [Sarukkai] and phonetic lattice methods [James] and combinations of the two [Yu]. In the first approach, speed is achieved by generating a description of the speech signal using a subset of sub-word descriptors. These descriptors are used to narrow the search space at retrieval time. In the second approach, the speech is indexed to produce a lattice of likely phonemes that can be searched quickly for any given phoneme sequence. In all of these methods, accuracy has been sacrificed for speed.

The Nexidia High-Speed Phonetic Search Engine

We now introduce another approach to phonetic searching, illustrated in Figure 1. This high-speed algorithm [Clements et al. 2001a; Clements et al. 2001b; Clements et al. 2007; U.S. patents 7,231,351; 7,263,484; 7,313,521; 7,324,939; 7,406,415] comprises two phases—indexing and searching. The first phase indexes the input speech to produce a phonetic search track and is performed only once. The second phase, performed whenever a search is needed for a word or phrase, is searching the phonetic search track. Once the indexing is completed, this search stage can be repeated for any number of queries. Since the search is phonetic, search queries do not need to be in any pre-defined dictionary, thus allowing searches for proper names, new words, misspelled words, jargon etc. Note that once indexing has been completed, the original media are not involved at all during searching and the search track could be generated on the highest-quality media available for improved accuracy (for example: μ-law audio for telephony), but then the audio could be replaced by a compressed representation for storage and subsequent playback (for example: GSM) afterwards.

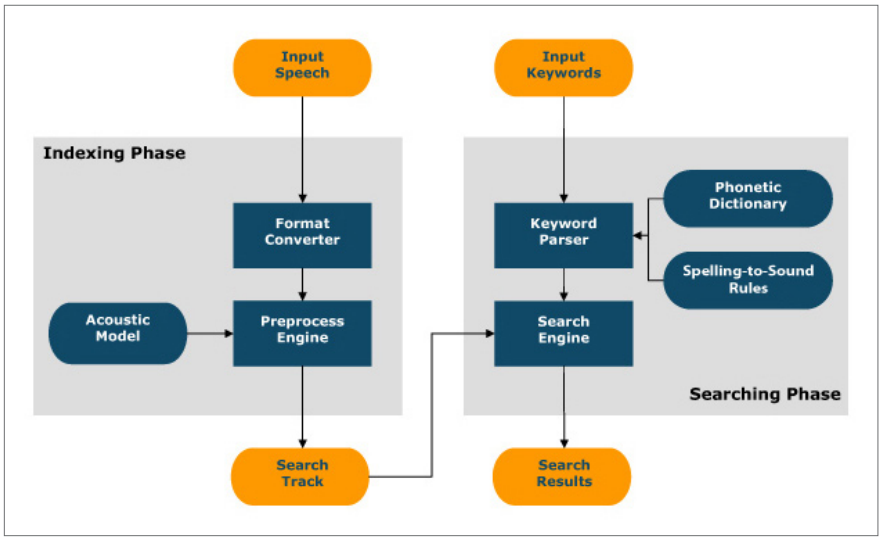

Figure 1
Nexidia High-Speed Phonetic Search Architecture

Indexing and the acoustic model

The indexing phase begins with format conversion of the input media (whose format might be MP3, ADPCM, QuickTime, etc.) into a standard audio representation for subsequent handling (PCM). Then, using an acoustic model, the indexing engine scans the input speech and produces the corresponding phonetic search track. An acoustic model jointly represents characteristics of both an acoustic channel (an environment in which the speech was uttered and a transducer through which it was recorded) and a natural language (in which human beings expressed the input speech). Audio channel characteristics include: frequency response, background noise and reverberation. Characteristics of a natural language include gender, dialect and accent of the speaker.

Nexidia typically produces two acoustic models for each language:

- a model for media with higher sampling rates, good signal-to-noise ratios, and more formal, rehearsed speech; and
- a model for media from a commercial telephony network, either landline or cellular handset, optimized for the more spontaneous, conversational speech of telephone calls.

Nexidia supports more than 30 languages including:

- Dutch
- English (North American, UK, and Australian)
- French (European, Canadian)
- Hindi
- German
- Japanese
- Korean
- Mandarin
- Russian
- Spanish (Latin American, Castilian)
- Thai

Additional languages are constantly in development. It typically takes less than a few weeks to develop a language pack for a new language.

Phonetic search track

The end result of phonetic indexing of an audio file is creating a Phonetic Audio Track (PAT file)—a highly compressed representation of the phonetic content of the input speech. Unlike LVCSR, whose essential purpose is to make irreversible (and possibly incorrect) bindings between speech sounds and specific words, phonetic indexing merely infers the likelihood of potential phonetic content as a reduced lattice, deferring decisions about word bindings to the subsequent searching phase.

PAT files are simple files that can be treated as metadata, associated and distributed with the originating media segments, produced in one environment, stored in data bases, transmitted via networks, and searched in another environment. The PAT file grows in size proportional to the length in time of the source media file, at around 3.7 MB per hour, equivalent to a bit rate of 8.6 kbps, i.e., 2/3 the rate of GSM telephony audio (13 kbps) or 1/15 the rate of typical MP3 (128 kbps).

KEYWORD PARSING

The searching phase begins with parsing the query string, which is specified as text containing one or more:

- words or phrases (e.g., “President” or “Supreme Court Justice”),
- phonetic strings (e.g., “_B _IY _T _UW _B _IY,” six phonemes representing the acronym “B2B”),
- temporal operators (e.g., “brain cancer &15 cell phone,” representing two phrases spoken within 15 seconds of each other).

A phonetic dictionary is referenced for each word within the query term to accommodate unusual words (whose pronunciations must be handled specially for the given natural language) as well as very common words (for which performance optimization is worthwhile). Any word not found in the dictionary is then processed by consulting a spelling-to-sound converter that generates likely phonetic representations given the word’s orthography.

SEARCH AND RESULTS LISTS

After words, phrases, phonetic strings and temporal operators within the query term are parsed, actual searching commences. Multiple PAT files can be scanned at high speed during a single search for likely phonetic sequences (possibly separated by offsets specified by temporal operators) that closely match corresponding strings of phonemes in the query term. Recall that PAT files encode potential sets of phonemes, not irreversible bindings to sounds. Thus, the matching algorithm is probabilistic and returns multiple results, each as a 4-tuple:

- PAT File (to identify the media segment associated with the putative hit)
- Start Time Offset (beginning of the query term within the media segment, accurate to one hundredth of a second)
- End Time Offset (approximate time offset for the end of the query term)
- Confidence Level (that the query term occurs as indicated, between 0.0 and 1.0)

Even during searching, irreversible decisions are postponed. Results are simply enumerated, sorted by confidence level, with the most likely candidates listed first. Postprocessing of the results list can be automated. Example strategies include hard thresholds (e.g., ignore results below 90% confidence), occurrence counting (e.g., a media segment gets a better score for every additional instance of the query term) and natural language processing (patterns of nearby words and phrases denoting semantics).

Typical web search engines strive to return multiple results on the first page so that the user can quickly identify one of the results as their desired choice. Similarly, an efficient user interface can be devised to sequence rapidly through a phonetic search results list, to listen briefly to each entry, to determine relevance and finally to select one or more utterances that meet specific criteria. Depending on available time and importance of the retrieval, the list can be perused as deeply as necessary.

STRUCTURED QUERIES

In addition to ad-hoc searches, Nexidia provides a more sophisticated technology to assist with contextual searches: a structured query. A structured query is similar to a finite state grammar that would be produced for an automatic speech recognition system. Examples of operators are AND, OR, and ANDNOT. Due to the special domain of search, several helpful extensions are also provided, such as attaching time windows to operators. Similar to Nexidia's standard ad-hoc phonetic search, both scores and time offsets are returned. By constructing complex queries, customers are able to easily generate document classifiers in addition to just detecting word or phrase occurrences. An example might be to identify how many calls in a call center's archive discuss problems with a rebate. Structured queries are simple to write and yet have the expressive power to capture complex Boolean and temporal relationships, as shown in the following example:

- Coupon = OR("coupon," "certificate," "rebate")
- Research = BEFORE_3("let me," OR("check," "do some research")))
- Problem = OR("I'm afraid," "unfortunately," Research)
- QUERY = AND_10(Coupon, Problem)

ADVANTAGES OF PHONETIC SEARCHING

The basic architecture of phonetic searching offers several key advantages over LVCSR and conventional word spotting:

- Speed, accuracy, scalability. The indexing phase devotes its limited time allotment only to categorizing input speech sounds into potential sets of phonemes—rather than making irreversible decisions about words. This approach preserves the possibility for high accuracy so that the searching phase can make better decisions when presented with specific query terms. Also, the architecture separates indexing and searching so that the indexing needs to be performed only once (typically during media ingest) and the relatively fast operation (searching) can be performed as often as necessary.

- Open vocabulary. LVCSR systems can only recognize words found in their lexicons. Many common query terms (such as specialized terminology and names of people, places and organizations) are typically omitted from these lexicons (partly to keep them small enough that LVCSRs can be executed cost effectively in real-time, and also because these kinds of query terms are notably unstable as new terminology and names are constantly evolving). Phonetic indexing is unconcerned about such linguistic issues, maintaining completely open vocabulary (or, perhaps more accurately, no vocabulary at all).

• Low penalty for new words. LVCSR lexicons can be updated with new terminology, names, and other words. However, this exacts a serious penalty in terms of cost of ownership—because the entire media archive must then be reprocessed through LVCSR to recognize the new words (an operation that typically executes only slightly faster than real time at best). Also, probabilities need to be assigned to the new words, either by guessing their frequency or context or by retraining a language model that includes the new words. The dictionary within the phonetic searching architecture, on the other hand, is consulted only during the searching phase, which is relatively fast compared to indexing. Adding new words incurs only another search, and it is often unnecessary to add words, since the spelling-to-sound engine can handle most cases automatically, or users can simply enter sound-it-out versions of words.

• Phonetic and inexact spelling. Proper names are particularly useful query terms—but also particularly difficult for LVCSR, not only because they may not occur in the lexicon as described above, but also because they often have multiple spellings (and any variant may be specified at search time). With phonetic searching, exact spelling is not required. For example, that mountainous region in NW Czechoslovakia can indeed be located by specifying "Sudetenland," but "Sue Dayton Land" will work as well. This advantage becomes clear with a name that can be spelled "Qaddafi," "Khaddafi," "Quadafy," "Kaddafi," or "Kadoffee"—any of which could be located by phonetic searching.

• User-determined depth of search. If a particular word or phrase is not spoken clearly, or if background noise interferes at that moment, then LVCSR will likely not recognize the sounds correctly. Once that decision is made, the correct interpretation is hopelessly lost to subsequent searches. Phonetic searching however returns multiple results, sorted by confidence level. The sounds at issue may not be the first (it may not even be in the top ten or 100), but it is very likely in the results list somewhere, particularly if some portion of the word or phrase is relatively unimpeded by channel artifacts. If enough time is available, and if the retrieval is sufficiently important, then a motivated user (aided by an efficient human interface) can drill as deeply as necessary. This capability is simply unavailable with LVCSR.

• Amenable to parallel execution. The phonetic searching architecture can take full advantage of any parallel processing accommodations. For example, a computer with dual processors can index twice as fast. Additionally, PAT files can be processed in parallel by banks of computers to search more media per unit time (or search tracks can be replicated in the same implementation to handle more queries over the same media).

CURRENT IMPLEMENTATION OF PHONETIC SEARCHING

Nexidia provides a range of product offerings to support the needs of a wide range of environments. The most basic form, called Nexidia Workbench, is a C++ toolkit that provides the basic functionality of indexing and searching on media file or data streams. The workbench requires users to develop their own end-to-end application. An extensive set of sample code is provided to assist users in quickly adding phonetic-based search functionality to their applications.

The Nexidia Enterprise Speech Intelligence (ESI) solution is a full server-hosted application that can be configured to automatically ingest media files from multiple sources, search for any number of user-defined term lists and queries, and analyze these results for statistical patterns. Initially designed for ease of integration into a commercial call center, Nexidia ESI allows call center operators to easily determine script compliance statistics, monitor topic trends over time, and drill down into call archives to the specific occurrence of desired events, all using an intuitive web interface.

Nexidia also offers Nexidia ESI Developers Edition (DE), a web services toolkit to allow a custom-built application to directly control and administer an ESI installation. Since the ESI DE toolkit uses web services, applications may be developed using virtually any development environment, such as Java, Visual Basic, etc.

Other Nexidia products include AudioFinder, a standalone desktop solution ideal for e-discovery in the legal market and audio forensics in general; Language Assessor, a web-based solution that automatically assesses the pronunciation and fluency of call center agent applicants; and a product suite designed for the rich media market that includes automatic tagging of video assets.

All Nexidia products are designed to provide high performance for both indexing and search. On a typical 3.0 GHz Dual Process Dual Core server, media files are indexed between 82 and 340 times faster than real-time. Once the PAT files are loaded from disk into memory (RAM), search speeds over 1.5 million times faster than real-time can be achieved (or equivalently, more than 400 hours of audio searched in a second). The engine is designed to take maximum advantage of a multi-processor system, such that a dual processor box achieves nearly double the throughput of a single processor configuration, with minimal overhead between processors. Compared to alternative LVCSR approaches, the Nexidia phonetic-based search engine provides a level of scalability not achievable by other systems.

The Nexidia engine comes with built-in support for a wide variety of common audio formats, including PCM, μ -law, A-law, ADPCM, MP3, QuickTime, WMA, g.723.1, g.729, g.726, Dialogic VOX, GSM and many others. Nexidia also provides a framework to support custom file-formats and devices, such as direct network feeds and proprietary codecs, through a provided plug-in architecture.

Performance of Nexidia Phonetic Search

There are three key performance characteristics of Nexidia's Phonetic Search: accuracy of results, index speed and search speed. All three are important when evaluating any audio search technology. This section will describe each of these in detail for the Nexidia phonetic-based engine.

RESULT ACCURACY

Phonetic-based search results are returned as a list of putative hit locations, in descending likelihood order. As a user progresses further down this list, they will find more and more instances of their query occurring. However, they will also eventually encounter an increasing amount of false alarms (results that do not correspond to the desired search term). This performance characteristic is best shown by a curve common in detection theory: the Receiver Operating Characteristic curve, or ROC curve, shown in Figure 2 and Figure 3.

To generate this curve, one needs experimental results from the search engine (the ordered list of putative hits) and the ideal results for the test set (acquired by manual review and documentation of the test data). For audio search, the ideal set is the verbatim transcripts of what was spoken in the audio. For a single query, first the number of actual occurrences in the ideal transcript is counted. The ROC curve begins at the 0,0 point on graph of False Alarms per Hour versus Probability of Detection. Results from the search engine are then examined, beginning from the top of the list. When a putative hit in the list matches the transcript, the detection rate increases, as the percentage of the true occurrences detected has just gone up (the curve

goes up). When the hit is not a match, the false alarm rate now increases (the curve now moves to the right). This continues until the false alarm rate reaches a pre-defined threshold. For any single query in generic speech, this curve normally has very few points, since the same phrase will only happen a few times, unless the same topic is being discussed over and over in the database. To produce a meaningful ROC curve, thousands of queries are tested with the results averaged together, generating smooth, and statistically significant, ROC curves.

There are two major characteristics that affect the probability of detection of any given query.

1 the type of audio being searched; and

2 the length and phoneme composition of the search terms themselves.

To address the first issue, Nexidia provides two language packs for each language, one designed to search broadcast-quality media and another for telephony-quality audio. The ROC curves for North American English in broadcast and telephony are shown in Figures 2 and 3 respectively.

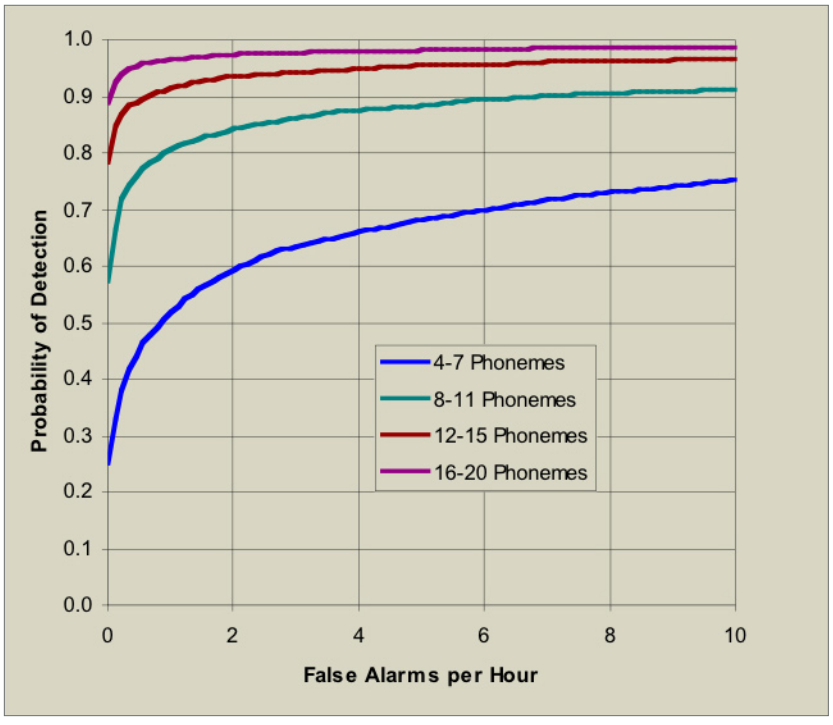

Figure 2

ROC Curves for the North American broadcast language pack

For example, using the North American English broadcast language pack and a query length of 12—15 phonemes, you can expect, on average, to find 85% of the true occurrences, with less than one false hit per 2 hours of media searched. The Nexidia engine provides an application the flexibility to choose to ensure a high probability of detection, by accepting results with a moderate confidence score, or to reduce false alarms (putative results will have a higher probability of being an actual desired result), by raising the score threshold and only accepting those with high confidence scores.

In a word-spotting system such as Nexidia, more phonemes in the query mean more discriminative information is available at search time. As shown by the four curves in the figures representing four different groups of query lengths, the difference can be dramatic. Fortunately, rather than short, single word queries (such as “no” or “the”), most real-world searches are for proper names, phrases, or other interesting speech that represent longer phoneme sequences. .

For the broadcast results, the test set is a ten-hour selection of ABC, CNN, and other newscasts, professionally transcribed and truthed by the Linguistic Data Consortium (LDC). For telephony, the test set is a 10-hour subset of the Switchboard and Switchboard Cellular corpora, also available from the LDC. Query terms were generated by enumerating all possible word and phrase sequences in the transcripts, and randomly choosing around ten thousand from this set.

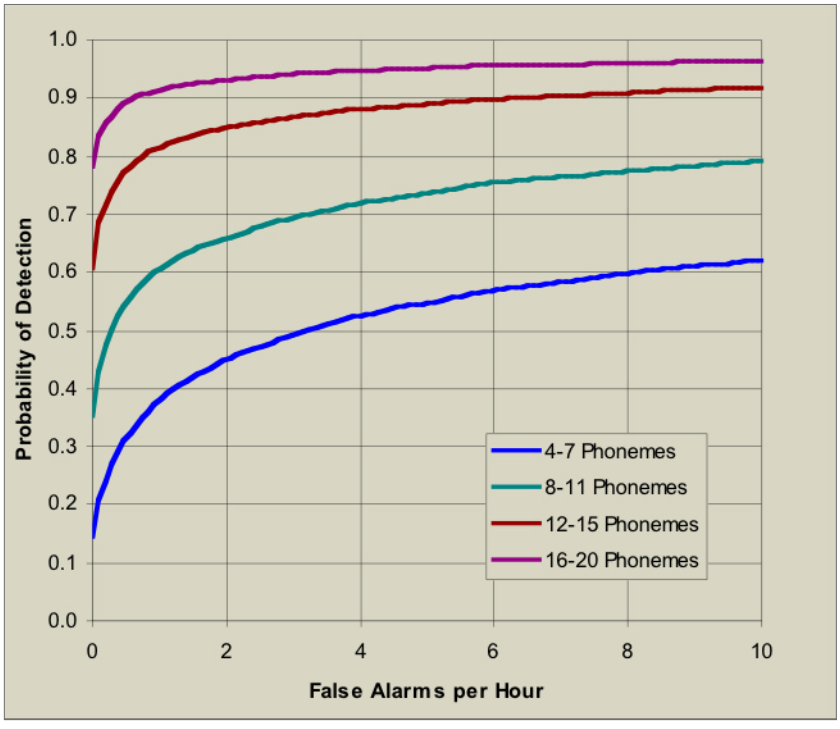

Figure 3

ROC Curves for the North American telephony language pack

INDEXING SPEED

Another significant metric of Nexidia's phonetic search is indexing speed (speed at which new media can be made searchable). This is a clear advantage for Nexidia, as the engine ingests media very rapidly. From call centers with hundreds of seats, media archives with tens of thousands of hours, or handheld devices with limited CPU and power resources, this speed is a primary concern, as this relates directly to infrastructure cost.

Indexing requires a relatively constant amount of computation per media hour, unless a particular audio segment is mostly silence, in which case the indexing rates are even greater. In the worst-case scenario of a call center or broadcast recording that contains mostly non-silence, ingest speeds for a server-class PC are given below in Table 1.

These speeds indicate that the indexing time for 1,000 hours of media is less than 1 hour of real time. Put another way, a single server at full capacity can index over 30,000 worth of media per day.

These results are for audio supplied in linear PCM or μ-law format to the indexing engine. If the audio is supplied in another format such as MP3, WMA, or GSM, there will be a small amount of format-dependent overhead to decode the compressed audio.

SEARCH SPEED

A final performance measure is the speed at which media can be searched once it has been indexed. Two main factors influence the speed of searching. The most important factor is whether the PAT files are in memory or on disk. Once an application

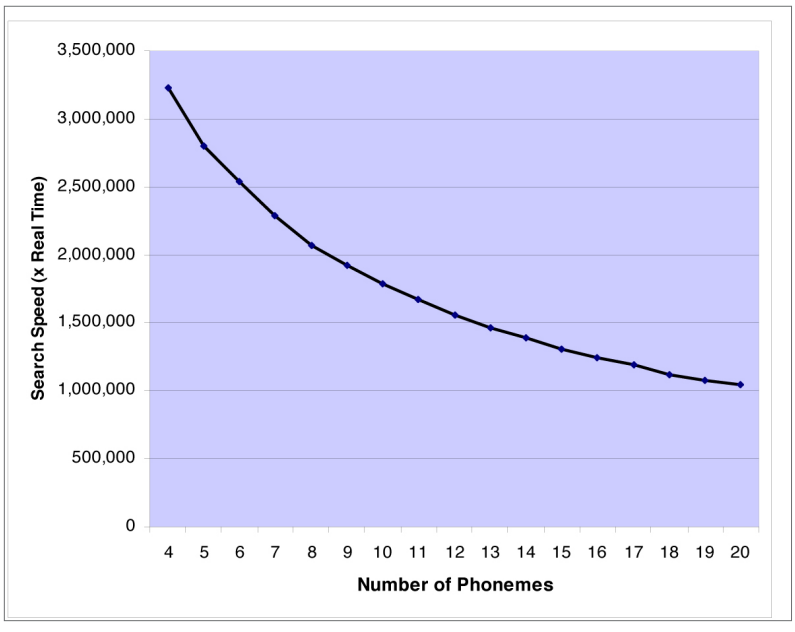

Figure 4

Search speed, in hours of media searched per second of CPU time, for a range of query lengths

requests a search track to be loaded (if it expects it to be needed soon), or else upon the first search of a track, the Nexidia search engine will load it into memory. Any subsequent searching will use this in-memory version, greatly speeding up when the same media is searched multiple times.

A second factor influencing search speed is the length, in phonemes, of the word or phrase in the query. Shorter queries run faster, as there are fewer calculations to make internal to the search engine.

Table 2 below shows the search speeds for a fairly average (12 phonemes long) query over a large set of in-memory PAT files, executed on a server-class PC.

Applications of Phonetic Search

The phonetic search technology presented in this paper has already found many applications and is applicable for many more. A brief summary of current and potential uses is:

- Call center data mining. Automatically search recorded archives in call-centers for useful and potentially profitable information, such as call trend analyses, finding problem areas in IVRs, fraud detection, and other uses.

INDEXING SPEED*	SERVER UTILIZATION
190	> 12.5% (single thread, only one CPU core used)
1,306	> 100% (8 threads, one thread per CPU core)

Table 1 **Indexing Speed (*in times faster than real time) Indexing speed on a 2-processor, 4-core server (Dell PowerEdge 2950, 2 x 3.16 GHz X5460 Quad Core, 4 GB RAM, 2 x 6 MB cache, 1.33 GHz FSB)**

SEARCH SPEED*	SERVER UTILIZATION
667,210	> 12.5% (single thread, only one CPU core used)
5,068,783	> 100% (8 threads, one thread per CPU core)

Table 2 **Search speed (* in times faster than real-time) for a 12-phoneme query on a 2-processor, 4-core server (Dell PowerEdge 2950, 2 x 3.16 GHz X5460 Quad Core, 4 GB RAM, 2 x 6 MB cache, 1.33 GHz FSB)**

- Call center quality control. Reduce expenses associated with manual oversight of CSR script compliance. Further, allow script compliance analysis across all calls, all seats of a center, rather than manual sampling of a small percentage.

- Searchable voice mail. Many people have begun using their email folders as their filing system, knowing text search can lead them to desired addresses, notes, discussions, and other information. Phonetic search now allows similar capabilities to voicemail.

- Real-time media research. Nexidia's phonetic search can run in "monitor" mode to return results with less than 1 second latency when monitoring up to 1,000 simultaneous audio streams per server.

- Archived media search. With Nexidia's phonetic search, it is possible first to find a podcast, lecture, or program of interest, and second, to immediately jump to the point in the recording talking about the desired topic.

- Notation, deposition and interview support. In scenarios where expensive transcription is not available, Nexidia's fast indexing and extremely fast search allows key remarks to be quickly found.

- Searchable notes for handheld users. Nexidia's search is lightweight enough to easily run on handheld devices. Quick notes no longer have to be written down—instead, one can just search the audio itself.

- Word or phrase detection. Lightweight search can easily power name-dialing, command-and-control, or other small applications currently handled by small speech recognition engines.

Conclusions

This paper has given an overview of the phonetic search technology developed at Nexidia. The method breaks searching into two stages: indexing and searching. The index stage happens only once per media file, and is extremely fast, at more than 1,000 faster than real-time on standard PC hardware. That file can then be searched independently any number of times, at a rate more than 5,000,000 times faster than real time. Search queries can be words, phrases, or even structured queries that allow operators such as AND, OR, and time constraints on groups of words. Search results are lists of time offsets into files, with an accompanying score giving the likelihood that a match to the query happened at this time.

Phonetic searching has several advantages over previous methods of searching audio media. By not constraining the pronunciation of searches, any proper name, slang, or even words that have been incorrectly spelled can be found, completely avoiding the out-of-vocabulary problems of speech recognition systems. Phonetic search is also fast. For deployments such as call centers with tens of thousands of hours of audio per day, the decision on selecting a subset to analyze need not be made, since with even modest resources all recordings can be indexed for search. Unlike other approaches, Nexidia's search technology is very scalable, allowing for fast and efficient searching and analysis of extremely large audio archives.

References

[Chang] E. I. Chang and R. P. Lippmann, "Improving Wordspotting Performance with Artificially Generated Data," in Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing, Atlanta, GA, Vol. 1, pp 283-286, 1996.

[Choi] J. Choi, D. Hindle, J. Hirshberg, I. Magrin-Chagnolleau, C. Kakatani, F. Pereira, A. Singhal, and S. Whittaker, "SCAN—Speech Content Based Audio Navigator: A Systems Overview," Proceedings Int. Conf. on Spoken Language Processing, 1998.

[Clements et al. 2001a] M. Clements, P. Cardillo, M. Miller, "Phonetic Searching of Digital Audio," NAB Broadcast Engineering Conference, Las Vegas, NV, April 2001.

[Clements et al. 2001b] M. Clements, P. Cardillo, M. Miller, "Phonetic Searching vs. LVCSR: How to Find What You Really Want in Audio Archives," AVIOS, San Jose, CA, April 2001.

[Clements et al. 2007] M. Clements and M. Gavalda, "Voice/Audio Information Retrieval: Minimizing the Need for Human Ears," Proceedings IEEE ASRU, Kyoto, Japan. December 2007.

[Garofolo] J. Garofolo, C. Auzanne, and E. Voorhees, "The TREC Spoken Document Retrieval Track: A Success Story," Proceedings of TREC-8, pp 107-116, Gaithersburg, MD, Nov. 1999.

[Graff] D. Graff, Z. Wu, R. McIntyre, and M. Liberman, "The 1996 Broadcast News Speech and Language-Model Corpus," Proceedings of the 1997 DARPA Speech Recognition Workshop, 1997.

[IBM] http://www-4.ibm.com/software/speech, ViaVoice®.

[James] D. A. James and S. J. Young, "A Fast Lattice-Based Approach to Vocabulary Independent Wordspotting," in Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing, Adelais, SA, Australia, Vol. 1, pp 377-380, 1994.

[Johnson] S.E. Johnson, P.C. Woodland, P. Jourlin, and K. Spärk Jones, "Spoken Document Retrieval for TREC-8 at Cambridge University," Proceedings of TREC-8, pp 197-206, Gaithersburg, MD, Nov. 1999.

[Jurafsky] D. Jurafsky and J. Martin, Speech and Language Processing, Prentice-Hall, 2000.

[Microsoft] X. Huang, A. Acero, F. Alleva, M. Hwang, L. Jiang, and M. Mahajan, Microsoft Windows Highly Intelligent Speech Recognizer: Whisper, Proceedings of ICASSP 95, volume 1, pp 93-97.

[Ng] K. Ng and V. Zue, "Phonetic Recognition for Spoken Document Retrieval," Proceedings of ICASSP 98, Seattle, WA, 1998.

[Philips] http://www.speech.be.philips.com, Speech Pearl®.

[Sarukkai] R. R. Sarukkai and D. H. Ballard, "Phonetic Set Indexing for Fast Lexical Access," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 20, no. 1, pp 78-82, January, 1998.

[Virage] http://www.virage.com, VideoLogger® and AudioLogger®.

[Wilpon] J. Wilpon, L. Rabiner, L. Lee, and E. Goldman, "Automatic Recognition of Keywords in Unconstrained Speech Using Hidden Markov Models," IEEE Transactions on Acoustics, Speech, and Signal Processing, Vol. 38, no. 11, pp 1870-1878, November, 1990.

[Wohlford] R. Wohlford, A. Smith, and M. Sambur, "The Enhancement of Wordspotting Techniques," in Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing, Denver, CO, Vol. 1, pp 209-212, 1980.

[Yu] P Yu, K. Chen, C. Ma, and F. Seide, "Vocabulary-Independent Indexing of Spontaneous Speech," IEEE Transactions on Speech and Audio Processing, volume 13, no. 5, September 2005.