Speech Recognition HOWTO

Stephen Cook

scook@gear21.com

Revision History

Revision v1.0v0.5	November 20, 2000September 13, 2000	Revised by: sccscc
Added info on L and HTKInitial HOWTO Submission		

Automatic Speech Recognition (ASR) on Linux is becoming easier. Several packages are available for users as well as developers. This document describes the basics of speech recognition and describes some of the available software.

Table of Contents

Table of Contents

1. Legal Notices

1.1. Copyright/License

This document is copyrighted (c) 2000 Stephen C. Cook.

LICENSE: This document may be reproduced and distributed in whole or in part, in any medium physical or electronic, provided that this license notice is displayed in the reproduction. Commercial redistribution is permitted and encouraged. Thirty days advance notice, via email to the author, of redistribution is appreciated, to give the author time to provide updated documents.

All modified documents, including translations, anthologies, and partial documents, must meet the following requirements:

- Modified versions must be labeled as such.
- The person making the modifications must be identified.
- Acknowledgement of the original author must be retained.
- The location of the original unmodified document be identified.
- The original author's name(s) may not be used to assert or imply endorsement of the resulting document without the original author's permission.
- The author be notified by email of the modification in advance of redistribution.
- As a special exception, anthologies of LDP documents may include a single copy of these license terms in a conspicuous location within the anthology and replace other copies of this license with a reference to the single copy of the license without the document being considered "modified" for the purposes of this section.

Mere aggregation of LDP documents with other documents or programs on the same media shall not cause this license to apply to those other works.

All translations, derivative documents, or modified documents that incorporate this document may not have more restrictive license terms than these, except that you may require distributors to make the resulting document available in source format.

1.2. Disclaimer

The author disclaims all warranties with regard to this document, including all implied warranties of merchantability and fitness for a certain purpose; in no event shall the author be liable for any special, indirect or consequential damages or any damages whatsoever resulting from loss of use, data or profits, whether in an action of contract, negligence or other tortious action, arising out of or in connection with the use of this document.

1.3. Trademarks

All trademarks contained in this document are copyright/trademark of their respective owners.

2. Forward

2.1. About This Document

This document is targeted at the beginner to intermediate level Linux user interested in learning about Speech Recognition and trying it out. It may also help the interested developer in explaining the basics of speech recognition programming.

I started this document when I began researching what speech recognition software and development libraries were available for Linux. Automated Speech Recognition (ASR or just SR) on Linux is just starting to come into its own, and I hope this document gives it a push in the right direction − by supporting both users and developers of ASR technology.

I have left a variety of SR techniques out of this document, and instead I have focused on the "HOWTO" aspect (since this is a howto...). I have included a Publications section so the interested reader can find books and articles on anything not covered here. This is not meant to be a definitive statement of ASR on Linux.

For the most recent version of this document, check the LDP archive, or go to:
http://www.gear21.com/speech/index.html.

2.2. Acknowledgements

I would like to thank the following people for the help, reviewing, and support of this document:

- Jessica Perry Hekman
- Geoff Wexler

2.3. Comments/Updates/Feedback

If you have any comments, suggestions, revisions, updates, or just want to chat about ASR, please send an email to me at scook@gear21.com.

2.4. ToDo

The following things are left "to do":

- Add descriptions in the Publications section.
- Add more books to the Publications section.
- Add more links with descriptions.
- Enhance the description of the ASR system steps
- Include descriptions of FFTs and Filters.
- Include descriptions of DSP principles.

2.5. Revision History

v0.1 first rough draft – August 2000

v0.5 final draft – September 2000

3. Introduction

3.1. Speech Recognition Basics

Speech recognition is the process by which a computer (or other type of machine) identifies spoken words. Basically, it means talking to your computer, AND having it correctly recognize what you are saying.

The following definitions are the basics needed for understanding speech recognition technology.

Utterance

An utterance is the vocalization (speaking) of a word or words that represent a single meaning to the computer. Utterances can be a single word, a few words, a sentence, or even multiple sentences.

Speaker Dependance

Speaker dependent systems are designed around a specific speaker. They generally are more accurate for the correct speaker, but much less accurate for other speakers. They assume the speaker will speak in a consistent voice and tempo. Speaker independent systems are designed for a variety of speakers. Adaptive systems usually start as speaker independent systems and utilize training techniques to adapt to the speaker to increase their recognition accuracy.

Vocabularies

Vocabularies (or dictionaries) are lists of words or utterances that can be recognized by the SR system. Generally, smaller vocabularies are easier for a computer to recognize, while larger vocabularies are more difficult. Unlike normal dictionaries, each entry doesn't have to be a single word. They can be as long as a sentence or two. Smaller vocabularies can have as few as 1 or 2 recognized utterances (e.g."Wake Up"), while very large vocabularies can have a hundred thousand or more!

Accuract

The ability of a recognizer can be examined by measuring its accuracy – or how well it recognizes utterances. This includes not only correctly identifying an utterance but also identifying if the spoken utterance is not in its vocabulary. Good ASR systems have an accuracy of 98% or more! The acceptable accuracy of a system really depends on the application.

Training

Some speech recognizers have the ability to adapt to a speaker. When the system has this ability, it may allow training to take place. An ASR system is trained by having the speaker repeat standard or common phrases and adjusting its comparison algorithms to match that particular speaker. Training a recognizer usually improves its accuracy.

Training can also be used by speakers that have difficulty speaking, or pronouncing certain words. As long as the speaker can consistently repeat an utterance, ASR systems with training should be able to adapt.

3.2. Types of Speech Recognition

Speech recognition systems can be separated in several different classes by describing what types of utterances they have the ability to recognize. These classes are based on the fact that one of the difficulties of ASR is the ability to determine when a speaker starts and finishes an utterance. Most packages can fit into more than one class, depending on which mode they're using.

Isolated Words

Isolated word recognizers usually require each utterance to have quiet (lack of an audio signal) on BOTH sides of the sample window. It doesn't mean that it accepts single words, but does require a single utterance at a time. Often, these systems have "Listen/Not−Listen" states, where they require the speaker to wait between utterances (usually doing processing during the pauses). Isolated Utterance might be a better name for this class.

Connected Words

Connect word systems (or more correctly 'connected utterances') are similar to Isolated words, but allow separate utterances to be 'run−together' with a minimal pause between them.

Continuous Speech

Continuous recognition is the next step. Recognizers with continuous speech capabilities are some of the most difficult to create because they must utilize special methods to determine utterance boundaries. Continuous speech recognizers allow users to speak almost naturally, while the computer determines the content. Basically, it's computer dictation.

Spontaneous Speech

There appears to be a variety of definitions for what spontaneous speech actually is. At a basic level, it can be thought of as speech that is natural sounding and not rehearsed. An ASR system with spontaneous speech ability should be able to handle a variety of natural speech features such as words being run together, "ums" and "ahs", and even slight stutters.

Voice Verification/Identification

Some ASR systems have the ability to identify specific users. This document doesn't cover verification or security systems.

3.3. Uses and Applications

Although any task that involves interfacing with a computer can potentially use ASR, the following applications are the most common right now.

Dictation

Dictation is the most common use for ASR systems today. This includes medical transcriptions, legal and business dictation, as well as general word processing. In some cases special vocabularies are used to increase the accuracy of the system.

Command and Control

ASR systems that are designed to perform functions and actions on the system are defined as Command and Control systems. Utterances like "Open Netscape" and "Start a new xterm" will do just that.

Telephony

Some PBX/Voice Mail systems allow callers to speak commands instead of pressing buttons to send specific tones.

Wearables

Because inputs are limited for wearable devices, speaking is a natural possibility.

Medical/Disabilities

Many people have difficulty typing due to physical limitations such as repetitive strain injuries (RSI), muscular dystrophy, and many others. For example, people with difficulty hearing could use a system connected to their telephone to convert the caller's speech to text.

Embedded Applications

Some newer cellular phones include C&C speech recognition that allow utterances such as "Call Home". This could be a major factor in the future of ASR and Linux. Why can't I talk to my television yet?

---

4. Hardware

4.1. Sound Cards

Because speech requires a relatively low bandwidth, just about any medium−high quality 16 bit sound card will get the job done. You must have sound enabled in your kernel, and you must have correct drivers installed. For more information on sound cards, please see "The Linux Sound HOWTO" available at: http://www.LinuxDoc.org/. Sound card quality often starts a heated discussion about their impact on accuracy and noise.

Sound cards with the 'cleanest' A/D (analog to digital) conversions are recommended, but most often the clarity of the digital sample is more dependent on the microphone quality and even more dependent on the environmental noise. Electrical "noise" from monitors, pci slots, hard−drives, etc. are usually nothing compared to audible noise from the computer fans, squeaking chairs, or heavy breathing.

Some ASR software packages may require a specific sound card. It's usually a good idea to stay away from specific hardware requirements, because it limits many of your possible future options and decisions. You'll have to weigh the benefits and costs if you are considering packages that require specific hardware to function properly.

4.2. Microphones

A quality microphone is key when utilizing ASR. In most cases, a desktop microphone just won't do the job. They tend to pick up more ambient noise that gives ASR programs a hard time.

Hand held microphones are also not the best choice as they can be cumbersome to pick up all the time. While they do limit the amount of ambient noise, they are most useful in applications that require changing speakers often, or when speaking to the recognizer isn't done frequently (when wearing a headset isn't an option).

The best choice, and by far the most common is the headset style. It allows the ambient noise to be minimized, while allowing you to have the microphone at the tip of your tongue all the time. Headsets are available without earphones and with earphones (mono or stereo). I recommend the stereo headphones, but it's just a matter of personal taste.

You can get excellent quality microphone headsets for between $25 $100. A good place to start looking is http://www.headphones.com or http://www.speechcontrol.com.

A quick note about levels: Don't forget to turn up your microphone volume. This can be done with a program such as XMixer or OSS Mixer and care should be used to avoid feedback noise. If the ASR software includes auto−adjustment programs, use them instead, as they are optimized for their particular recognition system.

4.3. Computers/Processors

ASR applications can be heavily dependent on processing speed. This is because a large amount of digital filtering and signal processing can take place in ASR.

As with just about any cpu intensive software, the faster the better. Also, the more memory the better. It's possible to do some SR with 100MHz and 16M RAM, but for fast processing (large dictionaries, complex recognition schemes, or high sample rates), you should shoot for a minimum of a 400MHz and 128M RAM. Because of the processing required, most software packages list their minimum requirements.

Using a cluster (Beowulf or otherwise) to perform massive recognition efforts hasn't yet been undertaken. If you know of any project underway, or in development please send me a note! scook@gear21.com

5. Speech Recognition Software

5.1. Free Software

Much of the free software listed here is available for download at:
http://sunsite.uio.no/pub/Linux/sound/apps/speech/

5.1.1. XVoice

XVoice is a dictation/continuous speech recognizer that can be used with a variety of XWindow applications. It allows user−defined macros. This is a fine program with a definite future. Once setup, it performs with adequate accuracy.

XVoice requires that you download and install IBM's (free) ViaVoice for Linux (See Commercial Section). It also requires the configuration of ViaVoice to work correctly. Additionally, Lesstif/Motif (libXm) is required. It is also important to note that because this program interacts with X windows, you must leave X resources open on your machine, so caution should be used if you use this on a networked or multi−user machine.

This software is primarily for users. An RPM is available.

HomePage: http://www.compapp.dcu.ie/~tdoris/Xvoice/ http://www.zachary.com/creemer/xvoice.html

Project: http://xvoice.sourceforge.net

Community: http://www.onelist.com/community/xvoice

5.1.2. CVoiceControl/kVoiceControl

CVoiceControl (which stands for Console Voice Control) started its life as KVoiceControl (KDE Voice Control). It is a basic speech recognition system that allows a user to execute Linux commands by using spoken commands. CVoiceControl replaces KVoiceControl.

The software includes a microphone level configuration utility, a vocabulary "model editor" for adding new commands and utterances, and the speech recognition system.

CVoiceControl is an excellent starting point for experienced users looking to get started in ASR. It is not the most user friendly, but once it has been trained correctly, it can be very helpful. Be sure to read the documentation while setting up.

This software is primarily for users.

Homepage: http://www.kiecza.de/daniel/linux/index.html

Documents: http://www.kiecza.de/daniel/linux/cvoicecontrol/index.html

5.1.3. Open Mind Speech

Started in late 1999, Open Mind Speech has changed names several times (was VoiceControl, then SpeechInput, and then FreeSpeech), and is now part of the "Open Mind Initiative". This is an open source project. Currently it isn't fully operational and is primarily for developers.

This software is primarily for developers.

Homepage: http://freespeech.sourceforge.net

5.1.4. GVoice

GVoice is a speech ASR library that uses IBM's ViaVoice (free) SDK to control Gtk/GNOME applications. It includes libraries for initialization, recognition engine, vocabulary manipulation, and panel control. Development on this has been idle for over a year.

This software is primarily for developers.

Homepage: http://www.cse.ogi.edu/~omega/gnome/gvoice/

5.1.5. ISIP

The Institute for Signal and Information Processing at Mississippi State University has made its speech recognition engine available. The toolkit includes a front−end, a decoder, and a training module. It's a functional toolkit.

This software is primarily for developers.

The toolkit (and more information about ISIP) is available at: http://www.isip.msstate.edu/project/speech/

5.1.6. CMU Sphinx

Sphinx originally started at CMU and has recently been released as open source. This is a fairly large program that includes a lot of tools and information. It is still "in development", but includes trainers, recognizers, acoustic models, language models, and some limited documentation.

This software is primarily for developers.

Homepage: http://www.speech.cs.cmu.edu/sphinx/Sphinx.html

Source: http://download.sourceforge.net/cmusphinx/sphinx2−0.1a.tar.gz

---

5.1.7. Ears

Although Ears isn't fully developed, it is a good starting point for programmers wishing to start in ASR.

This software is primarily for developers.

FTP site: ftp://svr−ftp.eng.cam.ac.uk/comp.speech/recognition/

5.1.8. NICO ANN Toolkit

The NICO Artificial Neural Network toolkit is a flexible back propagation neural network toolkit optimized for speech recognition applications.

This software is primarily for developers.

Its homepage: http://www.speech.kth.se/NICO/index.html

5.1.9. Myers' Hidden Markov Model Software

This software by Richard Myers is HMM algorithms written in C++ code. It provides an example and learning tool for HMM models described in the L. Rabiner book "Fundamentals of Speech Recognition".

This software is primarily for developers.

Information is available at: http://www.itl.atr.co.jp/comp.speech/Section6/Recognition/myers.hmm.html

5.1.10. Jialong He's Speech Recognition Research Tool

Although not originally written for Linux, this research tool can be compiled on Linux. It contains three different types of recognizers: DTW, Dynamic Hidden Markov Model, and a Continuous Density Hidden Markov Model. This is for research and development uses, as it is not a fully functional ASR system. The toolkit contains some very useful tools.

This software is primarily for developers.

More information is available at: http://www.itl.atr.co.jp/comp.speech/Section6/Recognition/jialong.html

5.1.11. More Free Software?

If you know of free software that isn't included in the above list, please send me a note at: scook@gear21.com. If you're in the mood, you can also send me where to get a copy of the software, and any impressions you may have about it. Thanks!

5.2. Commercial Software

5.2.1. IBM ViaVoice

IBM has made true on their promise to support Linux with their series of ViaVoice products for Linux, though the future of their SDKs aren't set in stone (their licensing agreement for developers isn't officially released as of this date − more to come).

Their commercial (not−free) product, IBM ViaVoice Dictation for Linux (available at http://www−4.ibm.com/software/speech/linux/dictation.html) performs very well, but has some sizeable system requirements compared to the more basic ASR systems (64M RAM and 233MHz Pentium). For the $59.95US price tag you also get an Andrea NC−8 microphone. It also allows multiple users (but I haven't tried it with multiple users, so if anyone has any experience please give me a shout). The package includes: documentation (PDF), Trainer, dictation system, and installation scripts. Support for additional Linux Distributions based on 2.2 kernels is also available in the latest release.

The ASR SDK is available for free, and includes IBM's SMAPI, grammar API, documentation, and a variety of sample programs. The ViaVoice Run Time Kit provides an ASR engine and data files for dictation functions, and user utilities. The ViaVoice Command & Control Run Time Kit includes the ASR engine and data files for command and control functions, and user utilities. The SDK and Kits require 128M RAM and a Linux 2.2 or better kernel)

The SDKs and Kits are available for free at: http://www−4.ibm.com/software/speech/dev/sdk_linux.html

---

5.2.2. Abbot/AbbotDemo

Abbot is a very large vocabulary, speaker independent ASR system. It was originally developed by the Connectionist Speech Group at Cambridge University. It was transferred (commercialized) to SoftSound. More information is available at: http://www.softsound.com.

AbbotDemo is a demonstration package of Abbot. This demo system has a vocabulary of about 5000 words and uses the connectionist/HMM continuous speech algorithm. This is a demonstration program with no source code.

5.2.3. Entropic

The fine people over at Entropic have been bought out by Micro$oft... Their products and support services have all but disappeared. Their support for HTK and ESPS/waves+ is gone, and their future is in the hands of M$. Their old website as http://www.entropic.com has more information.

K.K. Chin advised me that the original developers of the HTK (the Speech Vision and Robotic Group at Cambridge) are still providing support for it. There is also a "free" version available at: http://htk.eng.cam.ac.uk. Also note that Microsoft still owns the copyright to the current HTK code...

5.2.4. More Commercial Products

There are rumors of more commercial ASR products becoming available in the near future (including L&H). I talked with a couple of L&H representatives at Comdex 2000 (Vegas) and none of them could give me any information on a Linux release, or even if they planned on releasing any products for Linux. If you have any further information, please send any details to me at scook@gear21.com.

6. Inside Speech Recognition

6.1. How Recognizers Work

Recognition systems can be broken down into two main types. Pattern Recognition systems compare patterns to known/trained patterns to determine a match. Acoustic Phonetic systems use knowledge of the human body (speech production, and hearing) to compare speech features (phonetics such as vowel sounds). Most modern systems focus on the pattern recognition approach because it combines nicely with current computing techniques and tends to have higher accuracy.

Most recognizers can be broken down into the following steps:

1. Audio recording and Utterance detection
2. Pre−Filtering (pre−emphasis, normalization, banding, etc.)
3. Framing and Windowing (chopping the data into a usable format)
4. Filtering (further filtering of each window/frame/freq. band)
5. Comparison and Matching (recognizing the utterance)
6. Action (Perform function associated with the recognized pattern)

Although each step seems simple, each one can involve a multitude of different (and sometimes completely opposite) techniques.

(1) Audio/Utterance Recording: can be accomplished in a number of ways. Starting points can be found by comparing ambient audio levels (acoustic energy in some cases) with the sample just recorded. Endpoint detection is harder because speakers tend to leave "artifacts" including breathing/sighing,teeth chatters, and echoes.

(2) Pre−Filtering: is accomplished in a variety of ways, depending on other features of the recognition system. The most common methods are the "Bank−of−Filters" method which utilizes a series of audio filters to prepare the sample, and the Linear Predictive Coding method which uses a prediction function to calculate differences (errors). Different forms of spectral analysis are also used.

(3) Framing/Windowing involves separating the sample data into specific sizes. This is often rolled into step 2 or step 4. This step also involves preparing the sample boundaries for analysis (removing edge clicks, etc.)

(4) Additional Filtering is not always present. It is the final preparation for each window before comparison and matching. Often this consists of time alignment and normalization.

There are a huge number of techniques available for (5), Comparison and Matching. Most involve comparing the current window with known samples. There are methods that use Hidden Markov Models (HMM), frequency analysis, differential analysis, linear algebra techniques/shortcuts, spectral distortion, and time distortion methods. All these methods are used to generate a probability and accuracy match.

(6) Actions can be just about anything the developer wants. *GRIN*

6.2. Digital Audio Basics

Audio is inherently an analog phenomenon. Recording a digital sample is done by converting the analog signal from the microphone to an digital signal through the A/D converter in the sound card. When a microphone is operating, sound waves vibrate the magnetic element in the microphone, causing an electrical current to the sound card (think of a speaker working in reverse). Basically, the A/D converter records the value of the electrical voltage at specific intervals.

There are two important factors during this process. First is the "sample rate", or how often to record the voltage values. Second, is the "bits per sample", or how accurate the value is recorded. A third item is the number of channels (mono or stereo), but for most ASR applications mono is sufficient. Most applications use pre−set values for these parameters and user's shouldn't change them unless the documentation suggests it. Developers should experiment with different values to determine what works best with their algorithms.

So what is a good sample rate for ASR? Because speech is relatively low bandwidth (mostly between 100Hz−8kHz), 8000 samples/sec (8kHz) is sufficient for most basic ASR. But, some people prefer 16000 samples/sec (16kHz) because it provides more accurate high frequency information. If you have the processing power, use 16kHz. For most ASR applications, sampling rates higher than about 22kHz is a waste.

And what is a good value for "bits per sample"? 8 bits per sample will record values between 0 and 255, which means that the position of the microphone element is in one of 256 positions. 16 bits per sample divides the element position into 65536 possible values. Similar to sample rate, if you have enough processing power and memory, go with 16 bits per sample. For comparison, an audio Compact Disc is encoded with 16 bits per sample at about 44kHz.

The encoding format used should be simple − linear signed or unsigned. Using a U−Law/A−Law algorithm or some other compression scheme is usually not worth it, as it will cost you in computing power, and not gain you much.

7. Publications

If there is a publication that is not on this list, that you think should be, please send the information to me at: scook@gear21.com.

7.1. Books

- "Fundamentals of Speech Recognition". L. Rabiner & B. Juang. 1993. ISBN: 0130151572.
- "How to Build a Speech Recognition Application". B. Balentine, D. Morgan, and W. Meisel. 1999. ISBN: 0967127815.
- "Speech Recognition : Theory and C++ Implementation". C. Becchetti and L.P. Ricotti. 1999. ISBN: 0471977306.
- "Applied Speech Technology". A. Syrdal, R. Bennett, S. Greenspan. 1994. ISBN: 0849394562.
- "Speech Recognition : The Complete Practical Reference Guide". P. Foster, T. Schalk. 1993. ISBN: 0936648392.
- "Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics and Speech Recognition". D. Jurafsky, J. Martin. 2000. ISBN: 0130950696.
- "Discrete−Time Processing of Speech Signals (IEEE Press Classic Reissue)". J. Deller, J. Hansen, J. Proakis. 1999. ISBN: 0780353862.
- "Statistical Methods for Speech Recognition (Language, Speech, and Communication)". F. Jelinek. 1999. ISBN: 0262100665.
- "Digital Processing of Speech Signals" L. Rabiner, R. Schafer. 1978. ISBN: 0132136031
- "Foundations of Statistical Natural Language Processing". C. Manning, H. Schutze. 1999. ISBN: 0262133601.

For a very LARGE online biography, check the Institut Fur Phonetik: http://www.informatik.uni−frankfurt.de/~ifb/bib_engl.html

7.2. Internet

news:comp.speech

Newsgroup dedicated to computer and speech.

- ♦ US: http://www.speech.cs.cmu.edu/comp.speech/
- ♦ UK: http://svr−www.eng.cam.ac.uk/comp.speech/
- ♦ Aus: http://www.speech.su.oz.au/comp.speech/

news:comp.speech.users

Newsgroup dedicated to users of speech software.

- ♦ http://www.speechtechnology.com/users/comp.speech.users.html

news:comp.speech.research

Newsgroup dedicated to speech software and hardware research.

news:comp.dsp

Newsgroup dedicated to digital signal processing.

news:alt.sci.physics.acoustics

Newsgroup dedicated to the physics of sound.

DDLinux Email List

Speech Recognition on Linux Mailing List.

- Homepage: http://leb.net/ddlinux/
- Archives: http://leb.net/pipermail/ddlinux/

Linux Software Repository for speech applications

http://sunsite.uio.no/pub/linux/sound/apps/speech/

Russ Wilcox's List of Speech Recognition Links

(excellent) http://www.tiac.net/users/rwilcox/speech.html

Online Bibliography

Online Bibliography of Phonetics and Speech Technology Publications. http://www.informatik.uni−frankfurt.de/~ifb/bib_engl.html

MIT's Spoken Language Systems Homepage

http://www.sls.lcs.mit.edu/sls/

Oregon Graduate Institute

Center for Spoken Language Understanding at Oregon Graduate Institute. An excellent location for developers and researchers. http://cslu.cse.ogi.edu/

IBM's ViaVoice Linux SDK

http://www−4.ibm.com/software/speech/dev/sdk_linux.html

Mississippi State

Mississippi State Institute for Signal and Information Processing homepage with a large amount of useful information for developers. http://www.isip.msstate.edu/projects/speech/

Speech Technology

ASR software and accessories. http://www.speechtechnology.com

Speech Control

Speech Controlled Computer Systems. Microphones, headsets, and wireless products for ASR. http://www.speechcontrol.com

Microphones.com

Microphones and accessories for ASR. http://www.microphones.com

21st Century Eloquence

"Speech Recognition Specialists." http://voicerecognition.com

Computing Out Loud

Primarily for Windows users, but good info. http://www.out−loud.com

Say I Can.com

"The Speech Recognition Information Source." http://www.sayican.com