Speech Recognition

LISTEN AND LEARN

IVAN ANASTASOVSKI

I am not a touch typist. Actually I am not a typist at all. For me, typing is a slow and laborious process involving two fingers, much frustration and mostly imperfect results. When I type I hit the keyboard hard. For punishment. All those among you feeling the same have much reason to rejoice - the dictation software is finally available for Linux.

IBM's ViaVoice dictation package for Linux comes with a special active microphone that provides for filtration of background noise. The software included on the CD also provides text to speech functionality. Unfortunately, JAVA runtime necessary for proper functioning of the software is not on the CD. What we need is JAVA runtime for Linux from the blackdown.org site. *http://www.blackdown.org/java-linux.html*. The version of Java this ViaVoice release has been tested with is JRE-1.2.2 revision RC4. I first downloaded JAVA runtime from Blackdown and installed it under */usr/local*

The next step is to change your path to include PATH to JAVA runtime (to make it permanent, change the appropriate bash configuration file)

```
# export PATH=/usr/local/jre1.2.2/bin:$PATH
```

Now we install IBM ViaVoice Software:

```
# mount /mnt/cdrom
# cd /mnt/cdrom
# ./vvsetup
```

There will be a licence agreement to read and click through, it will check for the presence of the Java runtime and it will, at the end of it all, give you your prompt back.

Unmount the CD-ROM.

```
# umount /mnt/cdrom
```

And on to the configuration - change to your normal user account, fix your PATH to stumble across your brand new JAVA runtime before finding any other hiding in the nooks and crannies of your

The userguru is used to start the software training session

vast hard drive (geeks dream too), and invoke:

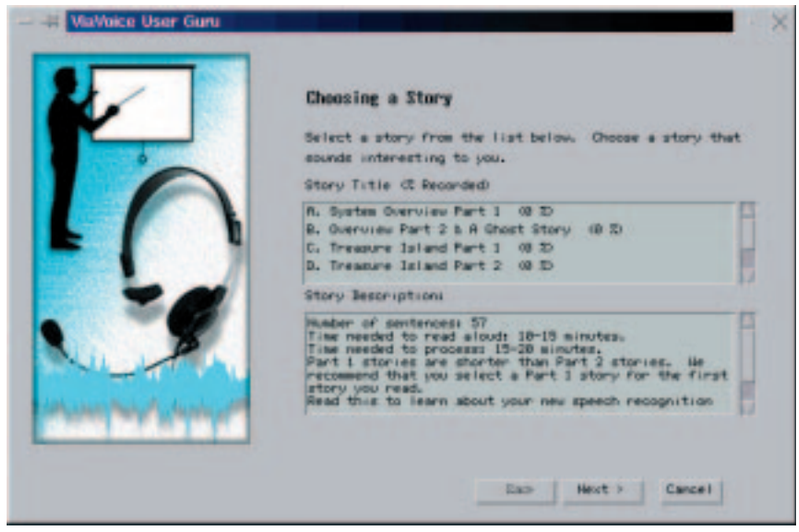
```
# vvstartuserguru
```

The first screen allows you to choose a *Profile Name* under which the training session will be saved. From there you go to the screen which gives rather precise instructions on just how to adjust your microphone for the best possible results.

The next screen allows you to test the playback capability of your sound card and to adjust its volume to a comfortable level. If all is not well, you need to get your sound card properly configured. Older cards are probably best configured using *#sndconfig* utility, while for the modern cards you are best off using the ALSA drivers.

Now we come to the often treacherous bit - microphone configuration. You click on the start button and read the given paragraph of text repeatedly until it beeps at you. If everything is OK it will beep very quickly and pronounce the quality of the sound excellent. The worst case is if it tells you that there was no output. Check your microphone connection to the sound card, keeping the wire away from any sources of electro-magnetic interference's and adjust the mixer settings. Right click on the microphone slider and make it the recording source, increase the microphone volume and gain and check if decreasing the volume of various other inputs lowers the line noise. If none of this works, it is very likely that your sound card is misconfigured. Look in the boxouts for examples.

Once you get your microphone to behave, you are ready to start a training session. Now just who is training whom is not entirely clear but the process should contribute to a long and fruitful future away from the keyboard (eventually). The result of this encounter is called Personal Voice Model. There are four texts to choose from - two short ones and two long ones, one of each being somewhat technical and the other one being somewhat literary. Choosing the long one results in better understanding between the parties involved. Being the only one who can press Ctrl+C and therefore having the upper hand (or the illusion of), I have chosen the short somewhat technical one. As you read it, the program highlights the parts corresponding to your vocalisation. If it gets stuck, it beeps at you, at which point you are meant to go back to the first non highlighted word and start again from there.



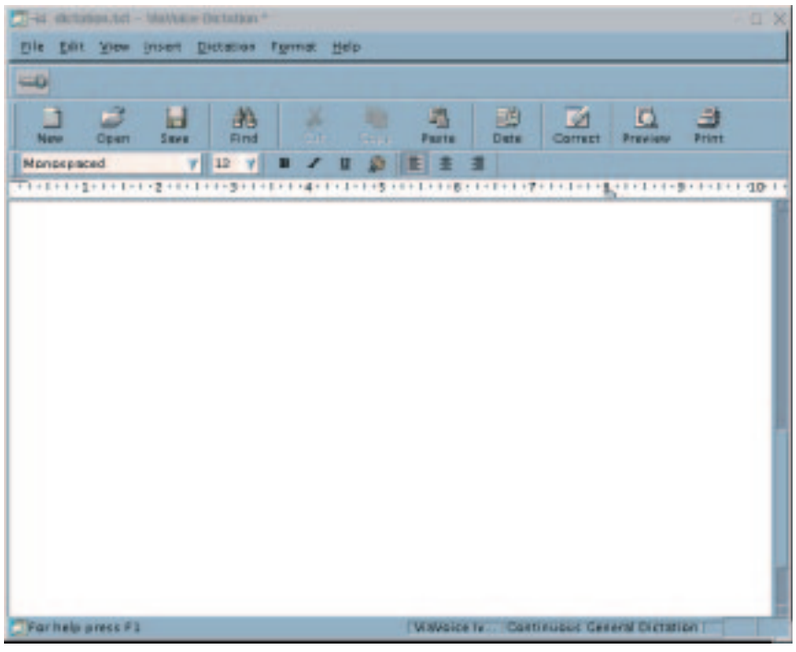
Choosing a story to train.

Once it has enough data it will go and play with the recording for a while (it estimates its playtime at 15 to 35 minutes - it actually took about three or four minutes on the test machine). When it comes back, it should be just familiar enough with your particular speech patterns to make dictation sessions accurate enough to keep you using it while it learns more about you and becomes better at its task.

To start a dictation session, we issue a command:

```
$ vvstartdictation
```

Adjusting the microphone settings



The word processor supplied

A window pops up. Among all the usual and expected window dressings, it contains an icon of a microphone. When it is green the window accepts the input from the microphone. As you speak, the words get recognised in short bursts of activity. It appears that the program does not work with single words, but with groups of words that it considers belong together. There are several ways to make the corrections - one of which is by highlighting and then speaking - quite often, even though you spoke only one word, it will still overwrite with several words which it thinks belong together. This presumably improves with usage. Another way to do the corrections is to use the voice commands to move about and edit. These commands together with commands to enter symbols are nicely grouped in a "What Can I Say" Window that can be displayed conveniently aside the main window. This is probably better left to the fortunate ones who are already good friends with this dictation software.

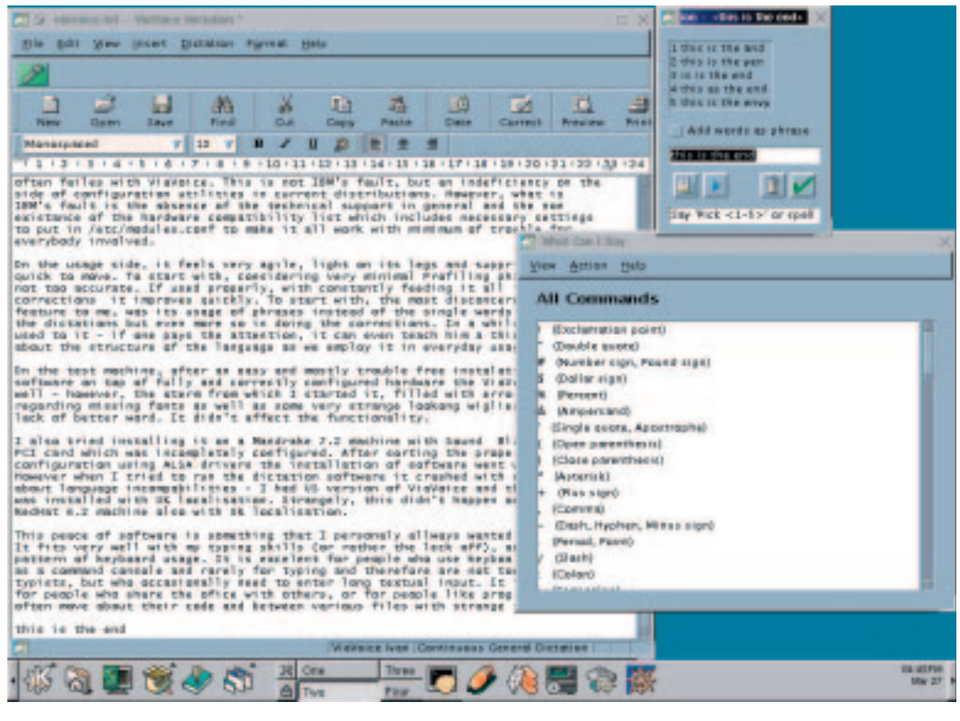
Another of the features is the Correction window. It allows the addition of phrases to a Personal Profile, thereby customising the software to the user and improving accuracy.

Dictation program can also do text-to-speech conversion. It can do it in one of two ways. It can play the highlighted text back to you in your own voice and in your own words exactly, or it can read to you its own take on what you dictated instead (in a voice somewhat resembling Stephen Hawking on an off day, and with an American accent). This feature is actually very useful - you can dictate a long peace from your bed, and even if there were a lot of mistakes in the captured text, you can still replay the original and do the corrections.

Conclusion:

Providing the hardware is properly configured, and one does not diverge significantly from the recommended versions of other necessary software, installation is fairly painless. This does not happen very often. The software uses features on the sound cards rarely used by other applications. Even when a sound card appears to be working correctly with other applications it still often fails with ViaVoice. This is not IBM's fault, but a deficiency on the side of configuration utilities in current distributions.

Finally all working



However, what is IBM's fault is the absence of the technical support in general and especially the non-existence of the hardware compatibility list that includes necessary settings to put in */etc/modules.conf* to make it all work with minimum of trouble for everybody involved.

On the usage side, it feels very agile, light on its legs and surprisingly quick to follow. To start with, considering very minimal Profiling phase it is not too accurate. If used properly, with constantly feeding it all the corrections it improves quickly. To start with, the most disconcerting feature to me, was its usage of phrases instead of the single words in taking the dictations but even more so in doing the corrections. In a while one gets used to it - if one pays the attention, it can even give him an insight or two into the structure of the language as we employ it in everyday usage.

On the test machine, after an easy and mostly trouble free installation of software on top of fully and correctly configured hardware the ViaVoice worked well. However, the xterm from which I started it, filled with error messages regarding missing fonts as well as some very strange looking symbols, for the lack of better word. It didn't affect the functionality.

I also tried installing it on a Mandrake 7.2 machine with SoundBlaster 128 PCI card which was incompletely configured. After sorting the proper configuration using ALSA drivers the installation and the profiling of software went well. However when I tried to run the dictation software it crashed with messages about language incompatibilities - I had US version of ViaVoice and the machine was installed with UK localisation. Strangely, this didn't happen on the Test RedHat 6.2 machine also with UK localisation.

This peace of software is something that I personally always wanted to have. It fits very well with my typing skills (or rather the lack off), and with my pattern of keyboard usage. It is excellent for people who use a keyboard mostly as a command console and rarely for typing and therefore are not touch typists, but still occasionally need to enter long textual input. It is not good for people who share the office with others, or for people like programmers who often move about their code between various files full of strange formatting. It is ideal for professionals like solicitors and doctors as well as for authors - one can lie in the bed and dictate for ours - as it doesn't feel to have the finality of putting things in black and white, it can do wonders for alleviating the writers block. It is probably most useful to the people with physical disabilities for obvious reasons.

In short, from me ViaVoice gets a thumb up - if it gets the hardware compatibility and configuration list it gets both thumbs up - I will be doing my dictations from my bed. ■

SoundBlaster 128 PCI configuration

I tried to install ViaVoice using this card on a machine with Mandrake 7.2, only to find that Mandrake installer configured the soundcard only partially causing it to fail. These settings worked for me allowing full installation and the initial training of ViaVoice. When I tried starting the dictation program itself, it failed complaining about wrong language - the test software is an early US version and the machine had an UK localisation. I am working on it :).

```
/etc/modules.conf:
alias char-major-116 snd
alias char-major-14 soundcore
alias snd-card-0 snd-card-ens1371
alias sound-slot-0 snd-card-0
alias sound-service-0-0 snd-mixer-oss
#alias sound-service-0-1 snd-seq-oss
alias sound-service-0-3 snd-pcm-oss
#alias sound-service-0-8 snd-seq-oss
alias sound-service-0-12 snd-pcm-oss
```

IBM installation requirements

Hardware:
Processor performance equivalent to Intel Pentium 233MHz with MMX with 256K L2 cache
128MB of RAM in total
140MB available hard disk space
Linux compatible 16 bit sound card of good recording quality (with microphone input jack)
CD-ROM drive
Andrea NC-61 Microphone or equivalent

Software:
Red Hat Linux, Version 6.2 with sound installed and enabled
Java, version JRE-1.2.2 revision RC4
http://www.blackdown.org/java-linux.html

Actual Test Hardware used:
Intel Pentium 650MHz with MMX with 256K L2 cache
128MB of RAM
SoundBlaster 16
CD-ROM drive
Andrea NC-61 Microphone {included}

Software:
Red Hat Linux, Version 6.2 with sound installed and enabled
Java, version JRE-1.2.2 revision RC4
http://www.blackdown.org/java-linux.html

Test System Sound Card Configuration

```
/etc/conf.modules(modules.conf):
alias sound-slot-0 sb
options sound dmabuf=1
alias midi opl3
options opl3 io=0x388
options sb io=0x220 irq=5 dma=0 dma16=5 mpu_io=0x330
```

The sound card I used is an old ISA SoundBlaster 16. All the settings are self explanatory except for dmabuf=1. ISA cards need the memory under 16MB for DMA access, and this setting reserves the required amount after the Kernel boots up and before other memory munching programs eat it all up