Setup
for automatic speech recognition experiments

These few pages provide information for the setting up for automatic speech recognition experiments. It focuses on the software and data to download and install.

Table of contents

The automatic speech recognition (ASR) experiments are based on the pocketsphinx speech recognition system. The examples of usage that are provided have been checked and run under linux (ubuntu), with python 3.

If your PC is running Windows 10, a simple solution consists in using the ubuntu distribution for Windows.

Another point to be aware of: In order to not alter your python installation when downloading new python packages, a solution consists in using a python virtual environment for the experiments (i.e. for downloading the python packages, for installing them, and for using them).

1 Setting up Ubuntu under windows 10

Relies on installing *Windows Subsystem for Linux*.

The web page https://docs.microsoft.com/en-us/windows/wsl/install-win10 specifies the installation process:

> Before installing any Linux distros for WSL, you must ensure that the "Windows Subsystem for Linux" optional feature is enabled:
>
> 1. Open PowerShell as Administrator and run:
>
> ```
> Enable-WindowsOptionalFeature -Online -FeatureName Microsoft-Windows-Subsystem-Linux
> ```
>
> 2. Restart your computer when prompted.

Install your Linux Distribution of Choice
Ubuntu :: https://www.microsoft.com/fr-fr/p/ubuntu/9nblggh4msv6?rtc=1#activetab=pivot:overviewtab

2 Directory for the ASR experiments

Let assume that your are creating a directory names ‘ASR-PocketSphinx’ on the desktop.
(of course, you can create the directory anywhere you want)

2.1 ubuntu under windows

If your directory ‘ASR-PocketSphinx’ is located on the windows desktop, the corresponding full paths are the following:

- under windows ➔ C:\Users\<WindowsUserName>\Desktop\ASR-PocketSphinx
- under unbuntu subsystem ➔ /mnt/c/Users/<WindowsUserName>/Desktop/ASR-PocketSphinx/

where ‘<WindowsUserName>’ is your username on Windows.

If you need to access the windows disks C or D, the paths under the ubuntu subsystem are:

- /mnt/c
- /mnt/d

2.2 Linux OS system

Standard naming of linux directories.

3 Python virtual environment

With python, one can define virtual environments, that is working environments that allows to install python packages without interfering with you standard python installation.

3.1 Creating python virtual environment

Go in the working directory you have created:

If under ubuntu windows subsystem

for example cd /mnt/c/Users/<WindowsUserName>/Desktop/ASR-PocketSphinx/
where ‘<WindowsUserName>’ is your username on Windows.

If needed, install the python virtual environment package (‘python3-venv’) under ubuntu

```
sudo apt-get update
sudo apt-get install python3-venv
```

Create python virtual environment, named ‘asr-env’:

python3 -m venv asr-env

This creates a sub directory named ‘asr-env’ in which the python packages will be installed.

3.2 How to work with a python virtual environment

Go in the working directory you have created:

If under ubuntu windows subsystem

for example cd /mnt/c/Users/<WindowsUserName>/Desktop/ASR-PocketSphinx/
where ‘<WindowsUserName>’ is your username on Windows.

Activate the virtual environment:

source asr-env/bin/activate

After this command, the python virtual environment is activated.

When it is activated, the linux prompt is modified, and starts with the name of the virtual environment in square brackets.

```
(asr-env)
…
… here you are working in the python virtual environment..
…
```

To exit from the python virtual environment, you just have to enter the following command:

deactivate

You should see the standard linux prompt.

4 Installing pocketsphinx package for python

4.1 Required ubuntu packages

The following ubuntu packages must be installed first:

```
sudo apt-get install swig
sudo apt-get install gcc
sudo apt-get install python3-dev
sudo apt-get install libpulse-dev
sudo apt-get install libasound2-dev
```

4.2 Package pocketsphinx pour python

If you plan to work in a python virtual environment, activate it first (see Section 3.2).

If needed, install or update the python installation package:

```
pip install wheel
```

Then install the pocketsphinx package

```
pip install pocketsphinx
```

5 Installing the ASR performance evaluation package

If you plan to work in a python virtual environment, activate it first (see Section 3.2).

Install the package

```
pip install asr-evaluation
```

6 Data for the ASR experiments

6.1 Dowload the data

Go in the working directory you have created

Download the data files from the UL ENT web site:

• **ps_data**	data for pocketsphinx (e.g., lexicons, language model, …)
• **ps_exemples**	some python programs using pocketsphins
• **td_corpus_digits**	speech data for the experiments – digits and digit sequences

The data are in zipped files.

Expand the content of the ps_data.zip file in a sub directory named ‘ps_data’.
Do similar processing for the other zip files (➔ sub directories ‘ps_exemples’ and ‘td_corpus_digits’).

6.2 Check that setup is correct

Go in the working directory you have created
If you plan to work in a python virtual environment, activate it first (see Section 3.2).

You should launch the following python programs, and check that the output is correct.
The python programs must be launched from the working directory.

1/ decoding with generic lexicon and ngram language model – version 1

Launching the program:

```
python ps_exemples/decoder_ngram.py
```

In the program output, you should see the following line:

```
Best hypothesis segments:  ['<s>', '<sil>', 'go', 'forward', 'ten', 'meters', '</s>']
```

2/ decoding with generic lexicon and ngram language model – version 2

Launching the program:

```
python ps_exemples/decoder_utt_ngram.py
```

In the program output, you should see the following line:

```
Best hypothesis segments:  ['<s>', '<sil>', 'go', 'forward', 'ten', 'meters', '</s>']
```

3/ decoding with the ‘turtle’ grammar – version 1

Launching the program:

```
python ps_exemples/decoder_jsgf.py
```

In the program output, you should see the following line:

```
Decoding with "turtle" language: go forward ten meters
```

And the line:

```
Decoding with "goforward" grammar: go forward ten meters
```

4/ decoding with the ‘turtle’ grammar – version 2

Launching the program:

```
python ps_exemples/decoder_utt_jsgf.py
```

In the program output, you should see the following line:

```
Decoding with "turtle" language: go forward ten meters
```

And the line:

```
Decoding with "goforward" grammar: go forward ten meters
```