GNU/Linux Administration - Support #604

Install Sirius on Ubuntu 14.04

04/21/2015 08:15 PM - Daniel Curtis

Status:	Suspended	**Start date:**	04/21/2015
Priority:	Normal	**Due date:**	
Assignee:	Daniel Curtis	**% Done:**	60%
Category:		**Estimated time:**	6.00 hours
Target version:	Ubuntu 14.04 / Derivative	**Spent time:**	6.00 hours

Description

This is a guide for installing the intelligent personal assistant Sirius on Ubuntu 14.04 minimal.

Prepare the environment

- Make sure the system is up to date:

```
sudo -s
apt-get update && apt-get upgrade
```

- Install wget:

```
apt-get install wget
```

Install the Sirius Application

NOTE: Sirius and its dependencies is several gigabytes, make sure to allocate enough storage space.

- Clone sirius from github:

```
git clone https://github.com/jhauswald/sirius.git
cd sirius/sirius-application
tar xzf question-answer.tar.gz
```

- Add additional repositories for ffmpeg

```
add-apt-repository ppa:kirillshkrogalev/ffmpeg-next
```

- Enable multiverse sources for libfaac-dev

```
apt-add-repository multiverse
```

- Update sources and install basic dependencies

```
apt-get update
apt-get install git zip unzip subversion sox default-jdk ant automake autoconf libtool bison l
ibboost-all-dev ffmpeg swig python-pip curl
```

- Install opencv dependencies

```
apt-get install build-essential checkinstall git cmake libfaac-dev libjack-jackd2-dev libmp3la
me-dev libopencore-amrnb-dev libopencore-amrwb-dev libsdl1.2-dev libtheora-dev libva-dev libvd
pau-dev libvorbis-dev libx11-dev libxfixes-dev libxvidcore-dev texi2html yasm zlib1g-dev
```

- Install tessaract text recognition

```
apt-get install tesseract-ocr tesseract-ocr-eng libtesseract-dev libleptonica-dev
```

- Install ATLAS library for Kaldi

```
apt-get install libatlas-dev libatlas-base-dev
```

- Install protobuf for image-matching

```
apt-get install libprotobuf-dev protobuf-compiler
```

- Install dependencies for the web application

```
pip install wtforms Flask requests pickledb
```

Install opencv

- Clone opencv from github

```
git clone https://github.com/Itseez/opencv.git opencv-2.4.9
cd opencv-2.4.9
git checkout 2.4.9
```

- Build and install opencv

```
mkdir build
cd build
cmake ..
make
make install
ldconfig -v
```

- Prepare kaldi

```
cd ~/sirius/sirius-application/speech-recognition/kaldi/scripts
./prepare.sh
```

- Compile Sirius

```
cd ~/sirius/sirius-application
./compile-sirius-servers.sh
```

Running Sirius

Automatic Speech Recognition (ASR)

Sirius supports three backends: Kaldi (DNN/HMM based), Pocketsphinx, and Sphinx4 (the latter are GMM/HMM based) to perform Automatic Speech Recognition.

- To open an ASR server:

```
cd ~/sirius/sirius-application/run-scripts
./start-asr-server.sh
```

  - or use the pocketsphinx ASR

```
./start-asr-server.sh pocketsphinx
```

  - or specify an ASR, hostname and port:

```
./start-asr-server.sh pocketsphinx localhost 8080
```

- In a separate terminal, test the ASR:

```
./sirius-asr-test.sh ../inputs/questions/what.is.the.speed.of.light.wav
```

Image Matching (IMM)

Image Matching uses SURF to match query images to a stored database.

- In image-matching/ first build and store a database of descriptors in protobuf format where the arguments are the name of the database and the directory containing the images:

```
cd ~/sirius/sirius-application/image-matching
./make-db.py landmarks matching/landmarks/db/
```

To change the database used by the IMM service, change the name in start-imm-server.py.

- In run-scripts/, open the IMM server:

```
cd ~/sirius/sirius-application/run-scripts
./start-imm-server.sh
```

- In a separate terminal, test IMM using:

```
./sirius-imm-test.sh ../image-matching/matching/landmarks/query/query.jpg
```

Question-Answering System (QA)

The Question-Answering system uses OpenEphyra and a Wikipedia database stored in Lemur's Indri format.

- Extract the Wikipedia database (after untaring and building question-answer):

```
cd ~/sirius/sirius-application
wget http://web.eecs.umich.edu/~jahausw/download/wiki_indri_index.tar.gz
tar xzvf wiki_indri_index.tar.gz -C question-answer/
```

- In run-scripts/, open the QA server:

```
./start-qa-server.sh
```

- In a separate terminal, test QA using:

```
./sirius-qa-test.sh "what is the speed of light"
```

Combining the Services

- It is very easy with Sirius to combine ASR and QA to create the full intelligent personal assistant pipeline. After opening multiple servers using ./start-<service>-server.sh, test an ASR-QA query using:

```
./sirius-asr-qa-test.sh ../inputs/real/what.is.the.capital.of.italy.wav
```

Resources

- http://sirius.clarity-lab.org/sirius/
- https://github.com/jhauswald/sirius

History

#1 - 04/21/2015 08:16 PM - Daniel Curtis

- Description updated

#2 - 04/22/2015 11:09 AM - Daniel Curtis

- Description updated

- Status changed from New to In Progress

- % Done changed from 0 to 20

#3 - 04/22/2015 06:23 PM - Daniel Curtis

- Description updated

- % Done changed from 20 to 40

#4 - 04/22/2015 08:30 PM - Daniel Curtis

- Description updated

#5 - 04/23/2015 07:30 AM - Daniel Curtis

- Description updated

- % Done changed from 40 to 50

#6 - 04/25/2015 05:08 AM - Daniel Curtis

- Description updated

- % Done changed from 50 to 60

#7 - 06/04/2017 09:47 PM - Daniel Curtis

- Status changed from In Progress to Suspended