Tahmina Tisha
Engineering notebook
Product vision
Learn about MAC Hardware
Mac installation

Kaldi Installation Journey Report on Mac Air 2020

Overview

This report documents the process and challenges encountered during the installation of the Kaldi speech recognition toolkit on a Mac Air 2020. The installation spanned from January 20 to January 23, 2024.

Initial Setup (January 20-22, 2024)

Cloning and Preparing the Environment

- Successfully cloned the necessary code and instructions from GitHub.
- Updated Visual Studio Code (VSCode) to ensure compatibility with the project requirements.
- Integrated GitHub with VSCode for streamlined code management.

Installing Homebrew and Connecting Dependencies

- Installed Homebrew, a package manager for macOS, which facilitated the installation of required dependencies for Kaldi.
- Followed the instructions provided by Dr. Liu, reaching Stage 2 of the installation process.

Installation Challenges (January 23, 2024)

Stage 1: Running Terminal Commands

Executed a series of commands to prepare for Kaldi installation:

- Navigated to the Kaldi directory and its subdirectories (**cd kaldi/tools/extras**).
- Ran **./check_dependencies.sh** to list all necessary packages.

Encountering and Overcoming macOS Specific Issues

- Faced limitations with macOS, as it does not support the **sudo** command for package installation.
- Initially attempted to install packages using **sudo apt install automake**, which resulted in errors and inappropriate Java installation prompts.
- Realized the necessity to use Homebrew (**brew**) for installing packages, due to macOS compatibility.

Installation of Key Packages (January 23, 2024)

Selection and Installation Process also stage 2 & 3

- A crucial step involved selecting and installing various packages necessary for running Kaldi effectively on macOS. These packages were identified through the **check_dependencies.sh** script.

Detailed List of Installed Packages

The following packages were installed, each serving a specific role in the setup of Kaldi:

1. **Automake**: Essential for generating Makefile templates compatible with various platform requirements.

2. **Autoconf**: Used for producing scripts that automatically configure software source code packages.
3. **Wget**: A network downloader to retrieve files from the web, crucial for fetching additional resources.
4. **Sox**: A command-line utility for converting, processing, and playing audio files.
5. **Gfortran**: The GNU Fortran compiler, is necessary for compiling Fortran dependencies.
6. **Subversion**: This version control system, commonly known as SVN, is crucial for managing and tracking changes in the Kaldi source code. Its reliable versioning capabilities are essential for a smooth installation and update process.
7. **Python 2.7**: Initially, Python 2.7 was installed as it was a prerequisite for some Kaldi scripts. However, compatibility issues with the latest macOS version posed significant challenges. Efforts to resolve this included experimenting with **python@2** and considering Python 3.7 as an alternative.
8. **Intel MKL**: Intel Math Kernel Library is critical for optimized mathematical computations, especially for tasks involving large data sets, which are common in speech recognition software like Kaldi. The installation and integration of Intel MKL on macOS required specialized handling due to system-specific configurations

Ongoing Efforts and Resolutions (January 23, 2024, Continued)

Addressing the Python 2.7 Challenge

- The incompatibility of Python 2.7 with the current macOS version presented a significant roadblock.
- Efforts were made to circumvent this by using **python@2** in the command line, which did not yield the desired result.
- Continued online research and troubleshooting to find a viable solution for integrating Python 2.7 or an alternative compatible with Kaldi.

Intel MKL Installation Complexities

- The installation of the Intel Math Kernel Library (MKL) posed a challenge due to limited prior knowledge and specific requirements for macOS.
- I explored various online forums and documentation to understand the best approach for installing and configuring Intel MKL on a Mac Air 2020.

Conclusion and Next Steps

As of January 23, 2024, the installation of Kaldi on a Mac Air 2020 remains in progress, with key focus areas being the resolution of Python 2.7 compatibility issues and the successful integration of Intel MKL. The journey so far has been both challenging and educational, providing valuable insights into software installation on macOS and the specific requirements of Kaldi.

Future Plans

- Continue researching and applying potential solutions for the Python 2.7 issue, possibly exploring alternative versions or workarounds that are compatible with both Kaldi and macOS.
- Seek assistance from DR Liu or my peers who have completed similar installations on macOS.

- Document the process and solutions for future reference and to assist others who may face similar challenges.

Report on Configuration and Installation Activities for Kaldi with Intel MKL and Python 2.7

Date: January 25, 2024

Objective: The main goal was to integrate the Intel Math Kernel Library (MKL) with the Kaldi speech recognition toolkit, and to ensure compatibility with Python 2.7 within the development environment, specifically VSCode.

Activities Undertaken:

1. **Initial Attempt to Download Intel MKL:**
    - Explored options for downloading the Intel MKL package, initially attempting to acquire it through an external website. The objective was to leverage Intel MKL's optimized mathematical routines to enhance matrix operations within Kaldi.
2. **Python 2.7 Installation:**
    - Successfully installed Python 2.7, specifically version 2.7.18, to maintain compatibility with certain Kaldi scripts that have dependencies on this Python version. The installation file for macOS was downloaded from the official Python website (Python 2.7.18 Release).
    - Installed Python within the Kaldi toolkit directory, specifically under **tools/extras**, to ensure seamless integration with Kaldi's build and configuration scripts.
3. **Dependency Checks and Configuration Attempts:**
    - Executed the **./check_dependencies.sh** script from the Kaldi root directory to verify system readiness and detect any missing dependencies. The script indicated that Intel MKL was not installed, prompting a need to download the installer package as per the instructions provided by the script.

Date: January 27, 2024

Continued Efforts to Install Intel MKL:

1. **Download and Installation Attempts:**
    - Continued efforts to install Intel MKL to enhance Kaldi's performance by enabling optimized mathematical computations. Faced challenges due to system compatibility and download source limitations.
2. **Script Execution Issues:**
    - Attempted to run **install_mkl.sh**, a script presumably designed to facilitate the Intel MKL installation process. However, encountered limitations as the script was primarily configured for Windows environments, leading to compatibility issues on the current system.

Summary: The installation and configuration process involved multiple steps, including successful installation of Python 2.7.18 within the Kaldi package to ensure script compatibility. Efforts to integrate Intel MKL encountered challenges, primarily due to system compatibility and the specificity of available installation scripts. Further actions may involve seeking alternative methods to install Intel MKL or consulting additional documentation to resolve compatibility issues.

Report on Mac Air M1 Compatibility with Kaldi and Intel MKL
Overview:
This brief explores the challenges faced when installing the Kaldi speech recognition toolkit and Intel Math Kernel Library (MKL) on a Mac Air M1, focusing on compatibility issues.
Kaldi Installation on Mac Air M1:
The journey to set up Kaldi on a Mac Air M1 from January 20 to January 23, 2024, highlighted several hurdles. Initial steps like cloning from GitHub and setting up VSCode went smoothly. Homebrew was essential for managing dependencies, a crucial step given macOS's unique ecosystem.
Challenges with macOS and MKL:

1. **macOS Quirks:** The macOS environment posed its own set of challenges, especially with its limited support for certain commands and the necessity to adapt by using Homebrew for package installations.
2. **Intel MKL Compatibility:** The crux of the matter lies with the Mac Air M1's ARM architecture. Intel MKL, designed primarily for Intel and AMD processors, encounters compatibility issues with ARM-based chips. The lack of native support for ARM in MKL complicates direct installation and optimal performance on the M1 chip.
3. **Python 2.7 Hurdles:** Compatibility issues with Python 2.7 further complicate the installation on macOS, especially considering the M1's cutting-edge architecture and the deprecated status of Python 2.7.

Concluding Thoughts:
The Mac Air M1, while a powerhouse in terms of performance and efficiency, presents unique challenges for software traditionally optimized for Intel/AMD architectures. The compatibility issues with Intel MKL stem from architectural differences, necessitating workarounds or alternatives for optimal functionality.
Next Steps:
Exploring ARM-compatible libraries or seeking updates from software developers might provide viable paths forward. Collaboration with peers and advisors like Dr. Liu could also shed light on alternative solutions or adjustments tailored to the M1's architecture.

Report: Intel MKL Installation and Alternative Libraries Exploration
Date: January 29th
Intel MKL Research and Installation Attempt

- Conducted thorough research on Intel Math Kernel Library (MKL) to understand its features and integration process.
- Attempted to follow the provided installation instructions for Intel MKL by executing the **./install_mkl.sh** script.
- Encountered a significant obstacle: the installation script returned an error indicating it is incompatible with the Darwin operating system. The exact error message was:
  ./install_mkl.sh: This script can be used on Linux only, and your system is Darwin.

Project File Configuration Adjustments

- Made adjustments to the project's configuration to align with the presumed Intel MKL installation path on a Darwin system:

    - Modified **AdditionalIncludeDirectories** and **AdditionalLibraryDirectories** to point to **/opt/intel/mkl/include** and **/opt/intel/mkl/lib** respectively, assuming a standard installation path for MKL on a Unix-like system.
    - Updated **AdditionalDependencies** to reference dynamic libraries with the **.dylib** extension, which is standard on macOS systems. Included libraries were **libmkl_rt.dylib**, **libmkl_intel_thread.dylib**, **libmkl_core.dylib**, and **libmkl_intel_lp64.dylib**.

Exploration of Alternative Libraries

- Given the challenges with Intel MKL installation on macOS, explored alternative libraries for numerical and scientific computing:
    - **NumPy**: Installed using the Python package manager pip with the command **pip install numpy**. NumPy provides support for large, multi-dimensional arrays and matrices, along with a collection of mathematical functions to operate on these arrays.
    - **SciPy**: Installed using pip with the command **pip install scipy**. SciPy builds on NumPy by adding a collection of algorithms and high-level commands for manipulation and visualization of data.
    - **OpenBLAS**: Investigated OpenBLAS, an open-source implementation of the BLAS library optimized for high performance on various CPU architectures. Installed on macOS using Homebrew with the command **brew install openblas**.

Conclusion

- The final attempt to verify the Intel MKL installation using **./check_dependencies.sh** confirmed that Intel MKL was not successfully installed, as indicated by the message: **Intel MKL does not seem to be installed.**

Future Plans

- Explore the option of setting up a virtual machine (VM) environment to facilitate the installation of Kaldi, which may include a compatible version of Intel MKL. Utilizing a VM could provide a controlled environment, potentially overcoming the compatibility issues encountered with the Darwin operating system.
- Research work for the MacOS including the reasons of the failed attempt of Kaldi Installation

Conflict: After wsl –install i was able to get Ubuntu running but after that terminal wasn't cooperating. User Manual wasnt clear about not being able to use sudo adduser <username>

Development Environment Setup: Summary and Action Plan

Setup Summary

February 1st:

Activities: Installed Parallels on Mac, set up Windows, downloaded and installed Visual Studio Code, and learned basics of PowerShell and Terminal.

Challenges: Difficulty in understanding and navigating Windows effectively.
February 4th:

Activities: Began detailed VS Code and Terminal setup, followed Adam's user manual for Kaldi installation on Ubuntu via WSL.
Challenges: Issues with terminal commands (sudo adduser <username>) post-WSL installation, attributed to unclear instructions in the Kaldi installation guide.
Challenges and Future Plans
Understanding Windows: Encountered initial challenges in navigating Windows, highlighting a need for targeted learning to improve efficiency in a Windows-based development environment.

Kaldi Installation Guide Clarity: Faced difficulties with terminal operations during Kaldi setup due to ambiguous instructions in the manual, particularly regarding user permissions and commands.

Future Plans:

Dedicate time to explore Windows-specific resources and tutorials to enhance familiarity and proficiency with the OS.
Collaborate with Adam to review and update the Kaldi installation manual, ensuring clear, step-by-step instructions are provided, especially for critical commands and WSL integration.

Powerpoint

1. **Kaldi Installation Challenges on Mac and Windows:**
    - Encountered macOS specific issues such as command limitations and Python 2.7 compatibility, leading to a switch to Windows via Parallels for a more compatible development environment.
2. **Learning Curve and Technical Hurdles:**
    - Faced initial challenges in understanding Windows operations and specific terminal commands for Kaldi installation, highlighting the need for targeted learning and clearer guidance.
3. **Future Plans for Resolution and Improvement:**
    - Plan to enhance familiarity with Windows, update the Kaldi installation guide for clarity, and seek community or peer support to overcome ongoing installation issues.

- **Initial Setup Challenges**: Faced difficulties with macOS limitations, necessitating the switch to Homebrew for package installations and addressing Python 2.7 incompatibility issues.
- **Intel MKL Installation Hurdles**: Encountered compatibility issues with Intel MKL on Mac Air M1's ARM architecture, requiring exploration of alternative libraries or solutions tailored to ARM.
- **Windows Environment Transition**: Installed Parallels and Windows on Mac to overcome compatibility issues, with a focus on learning Windows-specific operations for development.
- **Kaldi Installation Guide Improvements**: Plan to collaborate on refining the Kaldi installation manual, ensuring clarity in instructions, especially regarding WSL integration and terminal commands on Windows.
- **Future Learning and Collaboration**: Commit to enhancing understanding of Windows OS and working with peers to address installation challenges, aiming for a successful Kaldi setup.

Progress Report on Ubuntu Installation and Virtualization Efforts

February 5th

- Activity Summary: Initial work on setting up Ubuntu through Parallel Desktop Terminal was conducted. Successful execution led to Ubuntu being operational within this virtual environment.
- Future Plans: The next step involves the full utilization of Ubuntu within this setup to ensure all desired functionalities are accessible and running smoothly.

February 8th

- Challenges Encountered: Attempts to run Ubuntu via terminal and PowerShell led to multiple errors, indicating issues with compatibility or configuration.
- Future Plans: The primary objective is to conduct a thorough analysis of the encountered errors to understand their root causes and identify potential solutions.

February 10th

- Observation: Efforts to utilize Parallel Desktop Terminal for running Ubuntu reached a standstill, indicating a possible incompatibility or limitation within this environment for the desired use case.

February 13th

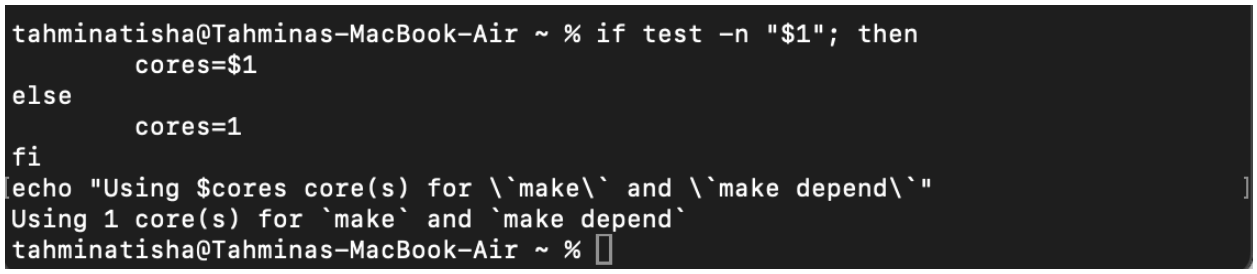
- New Direction: Transitioned to setting up VirtualBox as an alternative virtualization solution. The initial step involved downloading the Ubuntu ISO disk image in preparation for installation within VirtualBox.
- Next Steps: The focus will now be on installing and configuring Ubuntu within the VirtualBox environment, ensuring it meets the necessary requirements and resolves the challenges previously encountered with Parallel Desktop Terminal.

Feb 17th
Worked on user Manual 3.2.5

1.1.1 Installing on MAC

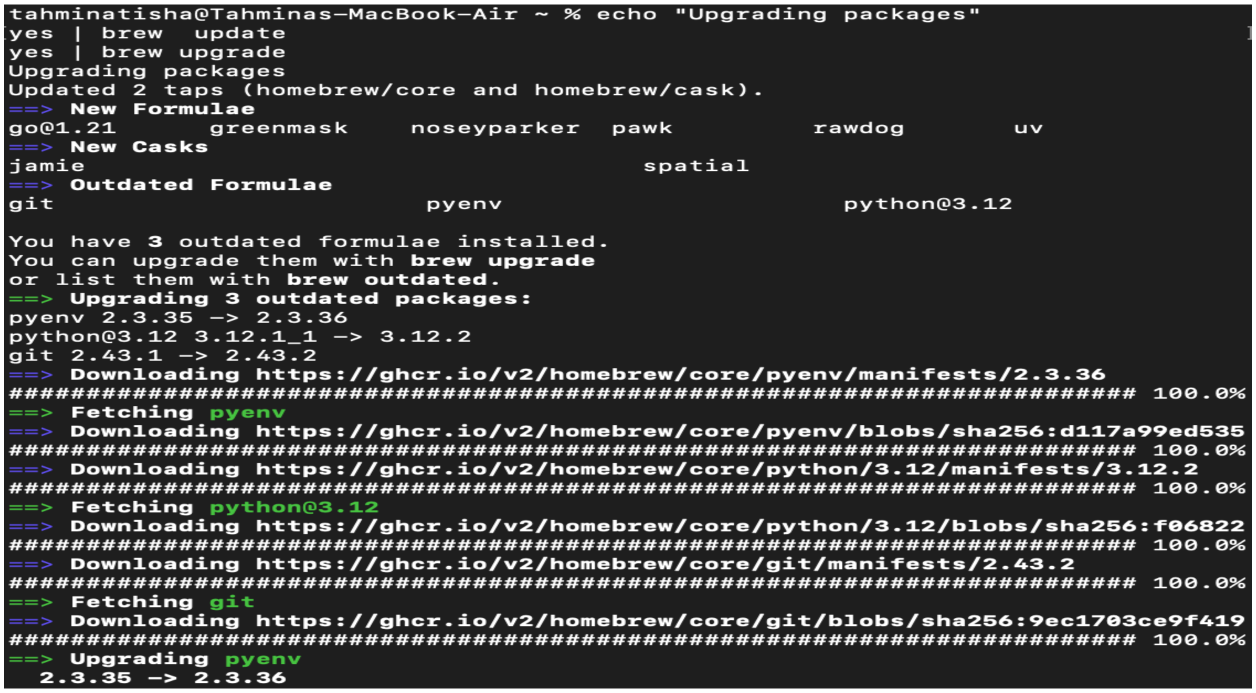
Cloning Kaldi Repository and Updating Visual Studio Code:
Begin by cloning the Kaldi speech recognition toolkit repository from GitHub onto your local machine. This involves downloading the source code and related files to your computer.
Ensure that Visual Studio Code (VSCode), a popular code editor, is updated to the latest version. This ensures compatibility with the Kaldi project and its requirements.
Integrate GitHub with VSCode for streamlined code management. This integration allows you to easily pull updates from the repository and manage version control directly within the VSCode interface.

You can use Mac Terminal to run the code.



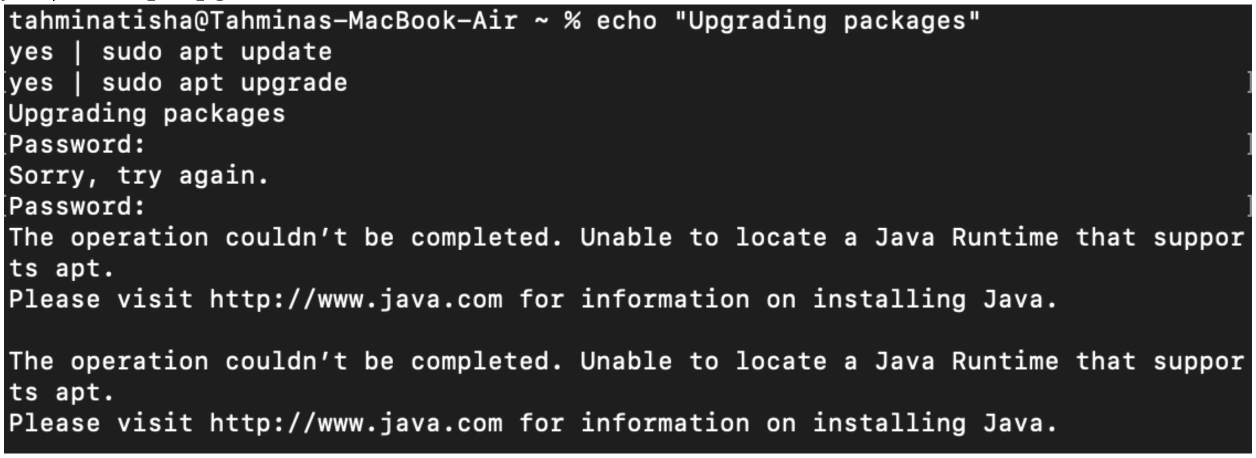
```
tahminatisha@Tahminas-MacBook-Air ~ % if test -n "$1"; then
        cores=$1
else
        cores=1
fi
echo "Using $cores core(s) for \`make\` and \`make depend\`"
Using 1 core(s) for `make` and `make depend`
tahminatisha@Tahminas-MacBook-Air ~ %
```

Installing Homebrew and Dependencies:
Homebrew is a package manager for macOS that simplifies the process of installing software and managing dependencies.
Install Homebrew on your Mac and follow the provided instructions for installing dependencies required for Kaldi. These dependencies may include libraries, compilers, and other tools necessary for building and running Kaldi successfully. https://brew.sh/



```
tahminatisha@Tahminas-MacBook-Air ~ % echo "Upgrading packages"
yes | brew  update
yes | brew upgrade
Upgrading packages
Updated 2 taps (homebrew/core and homebrew/cask).
==> New Formulae
go@1.21       greenmask    noseyparker  pawk         rawdog       uv
==> New Casks
jamie                                     spatial
==> Outdated Formulae
git                         pyenv                        python@3.12

You have 3 outdated formulae installed.
You can upgrade them with brew upgrade
or list them with brew outdated.
==> Upgrading 3 outdated packages:
pyenv 2.3.35 -> 2.3.36
python@3.12 3.12.1_1 -> 3.12.2
git 2.43.1 -> 2.43.2
==> Downloading https://ghcr.io/v2/homebrew/core/pyenv/manifests/2.3.36
######################################################################### 100.0%
==> Fetching pyenv
==> Downloading https://ghcr.io/v2/homebrew/core/pyenv/blobs/sha256:d117a99ed535
######################################################################### 100.0%
==> Downloading https://ghcr.io/v2/homebrew/core/python/3.12/manifests/3.12.2
######################################################################### 100.0%
==> Fetching python@3.12
==> Downloading https://ghcr.io/v2/homebrew/core/python/3.12/blobs/sha256:f06822
######################################################################### 100.0%
==> Downloading https://ghcr.io/v2/homebrew/core/git/manifests/2.43.2
######################################################################### 100.0%
==> Fetching git
==> Downloading https://ghcr.io/v2/homebrew/core/git/blobs/sha256:9ec1703ce9f419
######################################################################### 100.0%
==> Upgrading pyenv
  2.3.35 -> 2.3.36
```

Addressing macOS Limitations with Homebrew:
As you can see, In the instruction,
echo "Upgrading packages"
yes | sudo apt update
yes | sudo apt upgrade



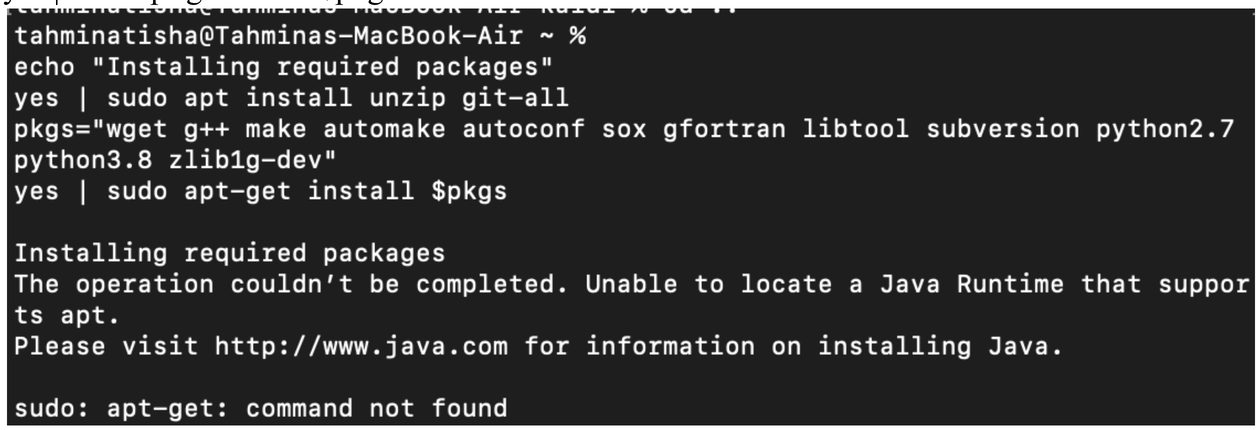
```
tahminatisha@Tahminas-MacBook-Air ~ % echo "Upgrading packages"
yes | sudo apt update
yes | sudo apt upgrade
Upgrading packages
Password:
Sorry, try again.
Password:
The operation couldn't be completed. Unable to locate a Java Runtime that suppor
ts apt.
Please visit http://www.java.com for information on installing Java.

The operation couldn't be completed. Unable to locate a Java Runtime that suppor
ts apt.
Please visit http://www.java.com for information on installing Java.
```

For the Kadi Installation, You have to use linux library to run the codes for the installation. However, due to the fact that MAC doesn't have linux, I had to change the sudo apt to brew update and brew upgrade for the installation for package.

macOS has certain limitations, such as restrictions on using the sudo command for package installations. Homebrew provides a workaround by allowing you to install packages without sudo privileges.
Utilize Homebrew to install key packages required for Kaldi, ensuring compatibility and ease of installation on macOS.

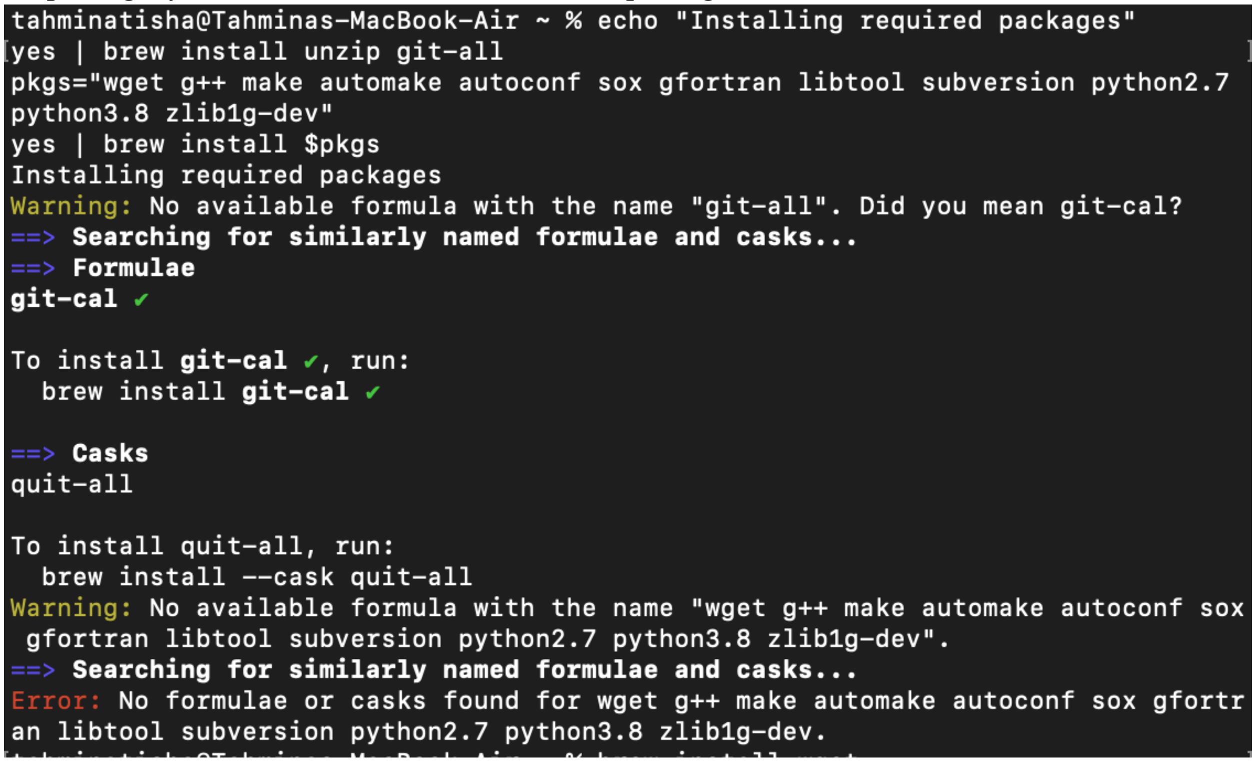
Installing Key Packages for Kaldi:
Install essential packages such as Automake, Autoconf, Wget, Sox, Gfortran, Subversion, Python 2.7, and Intel MKL. These packages are necessary for various aspects of building and running Kaldi, including compiling code, fetching resources, and performing mathematical computations.

```
echo "Installing required packages"
yes | sudo apt install unzip git-all
pkgs="wget g++ make automake autoconf sox gfortran libtool subversion python2.7 python3.8
zlib1g-dev"
yes | sudo apt-get install $pkgs
```



```
tahminatisha@Tahminas-MacBook-Air ~ %
echo "Installing required packages"
yes | sudo apt install unzip git-all
pkgs="wget g++ make automake autoconf sox gfortran libtool subversion python2.7
python3.8 zlib1g-dev"
yes | sudo apt-get install $pkgs

Installing required packages
The operation couldn't be completed. Unable to locate a Java Runtime that suppor
ts apt.
Please visit http://www.java.com for information on installing Java.

sudo: apt-get: command not found
```

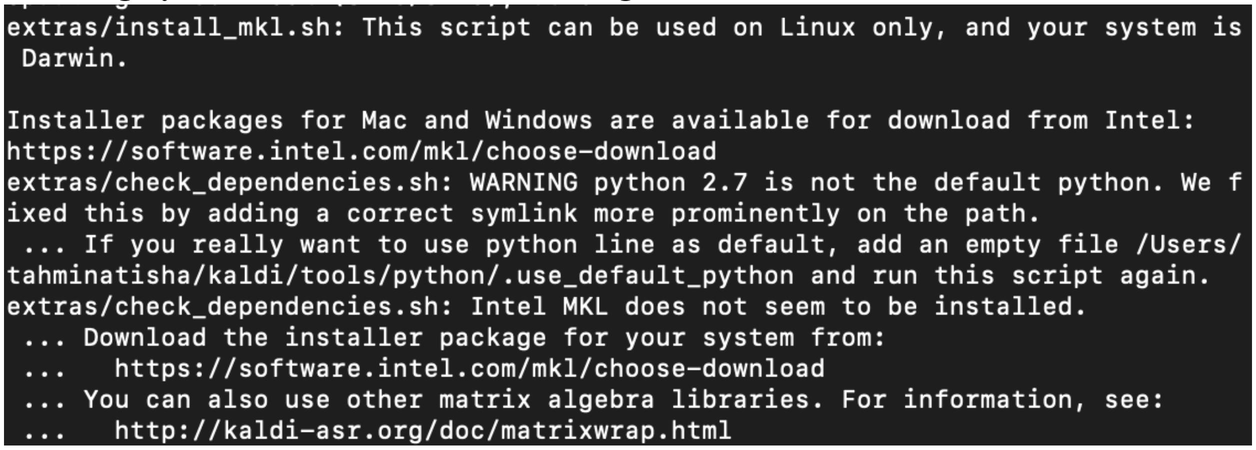
The installation of a package is very essential for the Kaldi installation. To be able to download the package, you have to use brew install for the packages on MAC OS.



```
tahminatisha@Tahminas-MacBook-Air ~ % echo "Installing required packages"
yes | brew install unzip git-all
pkgs="wget g++ make automake autoconf sox gfortran libtool subversion python2.7
python3.8 zlib1g-dev"
yes | brew install $pkgs
Installing required packages
Warning: No available formula with the name "git-all". Did you mean git-cal?
==> Searching for similarly named formulae and casks...
==> Formulae
git-cal ✔

To install git-cal ✔, run:
  brew install git-cal ✔

==> Casks
quit-all

To install quit-all, run:
  brew install --cask quit-all
Warning: No available formula with the name "wget g++ make automake autoconf sox
 gfortran libtool subversion python2.7 python3.8 zlib1g-dev".
==> Searching for similarly named formulae and casks...
Error: No formulae or casks found for wget g++ make automake autoconf sox gfortr
an libtool subversion python2.7 python3.8 zlib1g-dev.
```

The package installation failed due to problems with python 2.7 and Intel MKL installation compatibility on MAC OS. Intel MKL is the essential software package to run the Kaldi application.

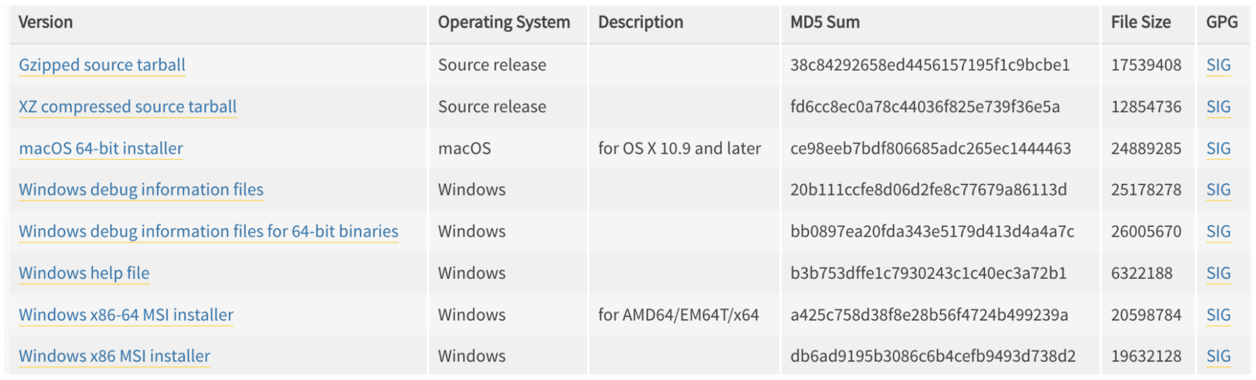
Resolving Python 2.7 and Intel MKL Challenges:



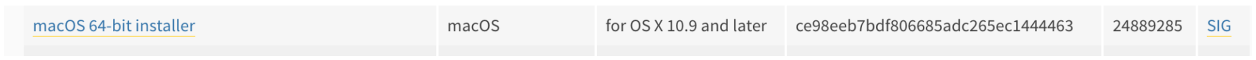
```
extras/install_mkl.sh: This script can be used on Linux only, and your system is
 Darwin.

Installer packages for Mac and Windows are available for download from Intel:
https://software.intel.com/mkl/choose-download
extras/check_dependencies.sh: WARNING python 2.7 is not the default python. We f
ixed this by adding a correct symlink more prominently on the path.
 ... If you really want to use python line as default, add an empty file /Users/
tahminatisha/kaldi/tools/python/.use_default_python and run this script again.
extras/check_dependencies.sh: Intel MKL does not seem to be installed.
 ... Download the installer package for your system from:
 ...   https://software.intel.com/mkl/choose-download
 ... You can also use other matrix algebra libraries. For information, see:
 ...   http://kaldi-asr.org/doc/matrixwrap.html
```

Python 2.7 compatibility issues may arise due to macOS updates or changes in software dependencies. Research and troubleshoot to find solutions that ensure compatibility with Python 2.7 while meeting the requirements of Kaldi.

However, this problem was resolved by downloading the python2.7.18 extertnally from the python.org. This resolved the problems for the python 2.7. Python 2.7.18 became an accepting alternate for the python 2.7.
https://www.python.org/downloads/release/python-2718/
This link will take you directly to python 2.7.18 download



Version	Operating System	Description	MD5 Sum	File Size	GPG
Gzipped source tarball	Source release		38c84292658ed4456157195f1c9bcbe1	17539408	SIG
XZ compressed source tarball	Source release		fd6cc8ec0a78c44036f825e739f36e5a	12854736	SIG
macOS 64-bit installer	macOS	for OS X 10.9 and later	ce98eeb7bdf806685adc265ec1444463	24889285	SIG
Windows debug information files	Windows		20b111ccfe8d06d2fe8c77679a86113d	25178278	SIG
Windows debug information files for 64-bit binaries	Windows		bb0897ea20fda343e5179d413d4a4a7c	26005670	SIG
Windows help file	Windows		b3b753dffe1c7930243c1c40ec3a72b1	6322188	SIG
Windows x86-64 MSI installer	Windows	for AMD64/EM64T/x64	a425c758d38f8e28b56f4724b499239a	20598784	SIG
Windows x86 MSI installer	Windows		db6ad9195b3086c6b4cefb9493d738d2	19632128	SIG

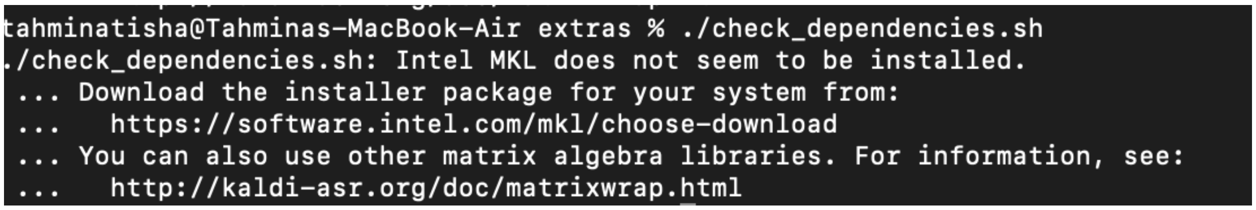


macOS 64-bit installer	macOS	for OS X 10.9 and later	ce98eeb7bdf806685adc265ec1444463	24889285	SIG

For MAC, I downloaded the MacOs 64-bits installer - This version is comaptitble for the kaldi installation and MacOs M1

Once, the downloading process is done,
# Section 4. Install tools for Kaldi

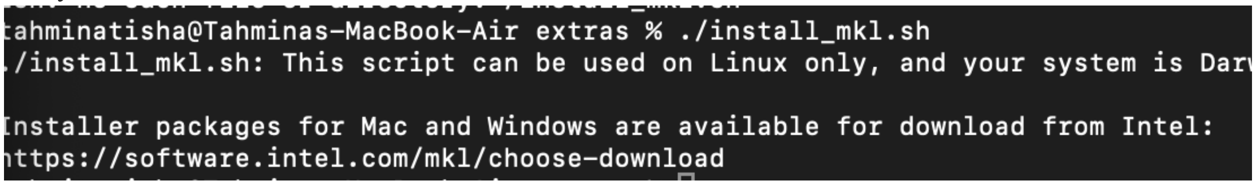
echo "Installing Kaldi"
echo "Installing Kaldi - Tools install"
git clone https://github.com/kaldi-asr/kaldi.git kaldi --origin upstream
cd ./kaldi/tools
extras/install_mkl.sh
extras/check_dependencies.sh



```
tahminatisha@Tahminas-MacBook-Air extras % ./check_dependencies.sh
./check_dependencies.sh: Intel MKL does not seem to be installed.
 ... Download the installer package for your system from:
 ...   https://software.intel.com/mkl/choose-download
 ... You can also use other matrix algebra libraries. For information, see:
 ...   http://kaldi-asr.org/doc/matrixwrap.html
```

Once all the package is downloaded, type
cd ./kaldi/tools
extras/install_mkl.sh
extras/check_dependencies.sh

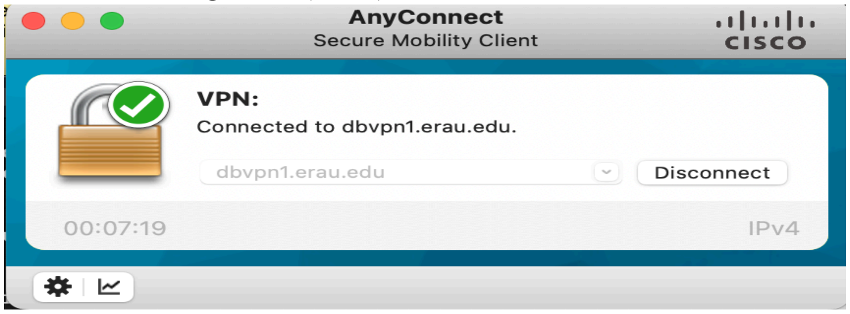
This will show all the packages installed for the installation. However, we can see that Intel MKL won't download on MacOS because the script can be used on linux only, and MacOs doesn't have linux and it runs on Darwin.
Darwin is based on the FreeBSD UNIX operating system, which differs slightly from GNU linux, so the syntax of some of the commands is different.



```
tahminatisha@Tahminas-MacBook-Air extras % ./install_mkl.sh
./install_mkl.sh: This script can be used on Linux only, and your system is Dar

Installer packages for Mac and Windows are available for download from Intel:
https://software.intel.com/mkl/choose-download
```

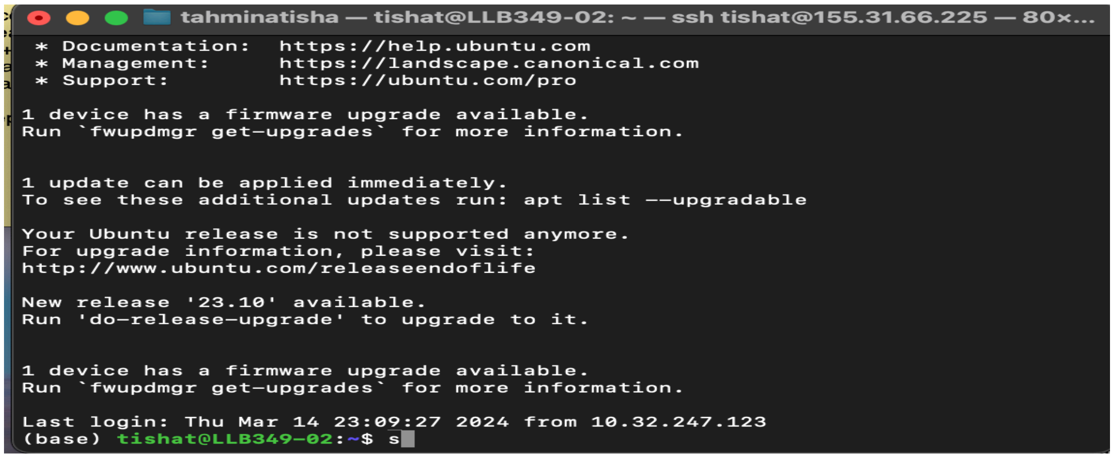
Intel Math Kernel Library (MKL) is crucial for optimized mathematical computations in Kaldi. However, installing and integrating MKL on macOS may present challenges due to system-specific configurations.

Installation using VPN (cisco)



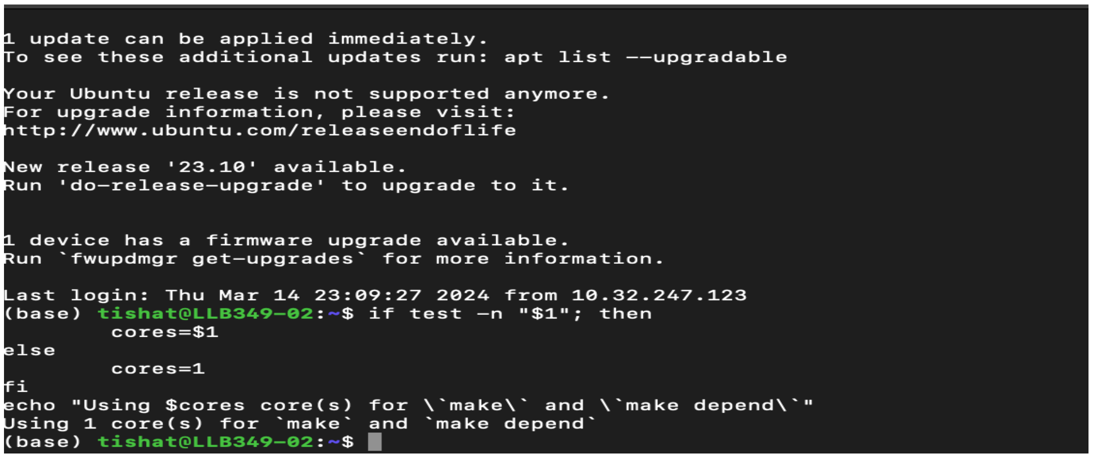
VPN:
Connected to dbvpn1.erau.edu.
dbvpn1.erau.edu
00:07:19

First ensure you are connected to the Embry Riddle Aeronautical University Daytona Beach VPN. This will allow us to use the computer that is connected to the campus. Currently I am connected to the computer that allows Ubuntu and has a sudo library.

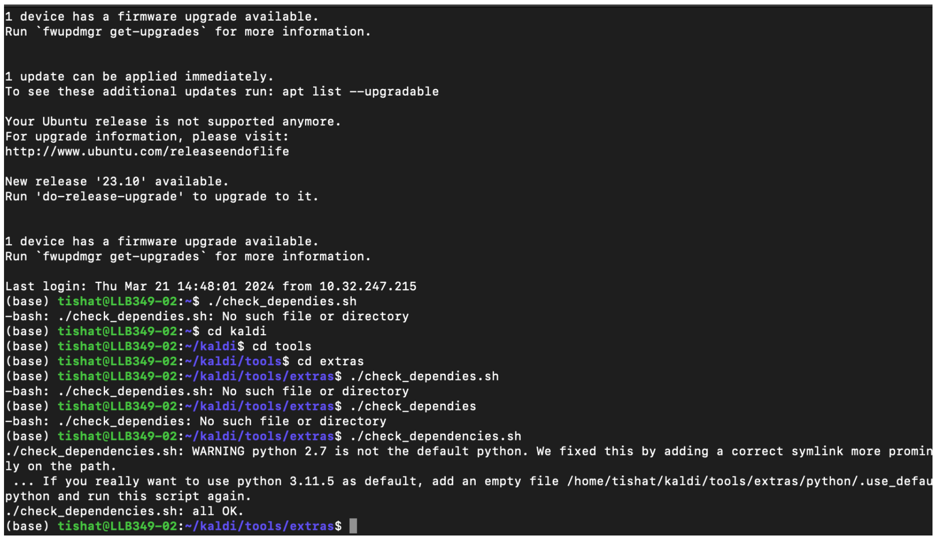


```
tahminatisha — tishat@LLB349-02: ~ — ssh tishat@155.31.66.225 — 80x...
 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/pro

1 device has a firmware upgrade available.
Run `fwupdmgr get-upgrades` for more information.


1 update can be applied immediately.
To see these additional updates run: apt list --upgradable

Your Ubuntu release is not supported anymore.
For upgrade information, please visit:
http://www.ubuntu.com/releaseendoflife

New release '23.10' available.
Run 'do-release-upgrade' to upgrade to it.


1 device has a firmware upgrade available.
Run `fwupdmgr get-upgrades` for more information.

Last login: Thu Mar 14 23:09:27 2024 from 10.32.247.123
(base) tishat@LLB349-02:~$ s
```

Once you are logged, you will be provided by the instruction to log into the sudo computers. You can set up the password.



```
1 update can be applied immediately.
To see these additional updates run: apt list --upgradable

Your Ubuntu release is not supported anymore.
For upgrade information, please visit:
http://www.ubuntu.com/releaseendoflife

New release '23.10' available.
Run 'do-release-upgrade' to upgrade to it.


1 device has a firmware upgrade available.
Run `fwupdmgr get-upgrades` for more information.

Last login: Thu Mar 14 23:09:27 2024 from 10.32.247.123
(base) tishat@LLB349-02:~$ if test -n "$1"; then
        cores=$1
else
        cores=1
fi
echo "Using $cores core(s) for \`make\` and \`make depend\`"
Using 1 core(s) for `make` and `make depend`
(base) tishat@LLB349-02:~$ 
```



```
1 device has a firmware upgrade available.
Run `fwupdmgr get-upgrades` for more information.


1 update can be applied immediately.
To see these additional updates run: apt list --upgradable

Your Ubuntu release is not supported anymore.
For upgrade information, please visit:
http://www.ubuntu.com/releaseendoflife

New release '23.10' available.
Run 'do-release-upgrade' to upgrade to it.


1 device has a firmware upgrade available.
Run `fwupdmgr get-upgrades` for more information.

Last login: Thu Mar 21 14:48:01 2024 from 10.32.247.215
(base) tishat@LLB349-02:~$ ./check_dependies.sh
-bash: ./check_dependies.sh: No such file or directory
(base) tishat@LLB349-02:~$ cd kaldi
(base) tishat@LLB349-02:~/kaldi$ cd tools
(base) tishat@LLB349-02:~/kaldi/tools$ cd extras
(base) tishat@LLB349-02:~/kaldi/tools/extras$ ./check_dependies.sh
-bash: ./check_dependies.sh: No such file or directory
(base) tishat@LLB349-02:~/kaldi/tools/extras$ ./check_dependies
-bash: ./check_dependies: No such file or directory
(base) tishat@LLB349-02:~/kaldi/tools/extras$ ./check_dependencies.sh
./check_dependencies.sh: WARNING python 2.7 is not the default python. We fixed this by adding a correct symlink more promin
ly on the path.
 ... If you really want to use python 3.11.5 as default, add an empty file /home/tishat/kaldi/tools/extras/python/.use_defau
python and run this script again.
./check_dependencies.sh: all OK.
(base) tishat@LLB349-02:~/kaldi/tools/extras$ 
```

1.1 Common Problems

While installing the Kaldi-ASR Toolkit you may encounter problems along the way. In this section it shall discuss common problems that you and others may have while installing the Kaldi Toolkit. These will each be broken up into sections.

3.3.1 Windows Subsystem for Linux

3.3.1.1 Timing Out Errors

If you are encountering timing out errors your screen may look like the following image below. [Paste Image]

If this is the case the first thing you should do is check your WiFi connection. The program pulls from git-hub which is on the internet and if you don't have a good connection the console may give you a Time Out Error. A good way to tell that you WiFi connection is good is by clicking on the WiFi symbol "[Paste Wifi Symbol here]", and checking to see if you are connected to a Router. The Wifi should have "good connection" or "connected" written next to the symbol and for maximum performance the WiFi symbol should be full bars.

If after this change it still doesn't work there may be another solution. That solution is that you may be running too many background tasks. Since Kaldi is a very big Program to install, running too many background tasks may interfere and slow the process down to the point where the system time's out. To check to see if you have too many background tasks running, you can check "Task Manager" or "Activity Monitor" on Mac. If you don't want to close out anything this way as you may not know what systems need to be running, you can also close out tabs that are pulling from the internet such as a search engine, video game digital distribution service, social media platform, ect. After resolving either solution you should see improvement to the speed of installation and you should not get any Timing Out Errors.

If Timing Out Errors still arise after working out each solution you can contact the Kaldi team by writing your problems on the Git-Hub. The link is provided here:

3.3.1.2 WSL doesn't have Sudo

This issue has not been solved yet

March 25, 2024
Installation

Operating Sy…	macOS
Processor	Apple M1
GPU	Apple (7-Core)
Total Installed…	16 GB
Display Size	13.3"
Resolution	2560 x 1600
Touchscreen	No
Total Installed…	256 GB
Inputs/Outputs	2 x USB-C (Thunderbolt 3 / USB4)
Wi-Fi	Wi-Fi 6 (802.11ax) with MU-MIMO Support
Bluetooth	5.0
Model Year	Late 2020
CPU	8-Core- Performance (4 Cores) Efficiency (4 Cores)
Graphics Type	Integrated
Memory Type	Embedded DRAM
Memory Confi..	16 GB (Onboard)
Panel Type	IPS-Type LCD
Aspect Ratio	16-10
Finish	Glossy
Maximum Bri…	400 nits / cd/m 2
Color Gamut	100% DCI-P3
Refresh Rate	60 Hz

- Apple M1 8-Core CPU
- 16GB Unified RAM | 256GB SSD
- 13.3" 2560 x 1600 Retina IPS Display
- 7-Core GPU | 16-Core Neural Engine
- Wi-Fi 6 (802.11ax) | Bluetooth 5.0
- 2 x Thunderbolt 3 / USB 4 Ports
- Backlit Magic Keyboard
- Force Touch Trackpad | Touch ID Sensor
- macOS
- Export a project for the web with iMovie up to 3x faster
- Integrate 3D effects into video in Final Cut Pro up to 5x faster
- For the first time, play back and edit multiple streams of full-quality, 4K ProRes video in Final Cut Pro without dropping a frame
- Export photos from Lightroom up to twice as fast

Operating Sy...	macOS
Processor	Apple M1
GPU	Apple (7-Core)
Total Installed...	16 GB
Display Size	13.3"
Resolution	2560 x 1600
Touchscreen	No
Total Installed...	256 GB

Summary coming soon