Ubuntu 18.04 VM for Computational Corpus Linguistics
Installation Notes

Setting up the Ubuntu VM

- download and install VirtualBox (http://www.virtualbox.org/)
- set up new VM for Linux Ubuntu 64-bit, variable size `.vdmk` disk image up to 42 GB
- add storage device: new optical device from disk image (Ubuntu installer DVD, downloaded from http://www.ubuntu.com/)
- basic Ubuntu installation: minimal install
  - user `issale`, password `issale`
  - when restarting to complete installation, installer DVD seems to be removed automatically

If you already have a regular Ubuntu installation, you can also follow the instructions below to provide a basic setup for computational corpus linguistics. You may need/want to skip some steps that are specific to a VBox VM (or to the ISSALE course).

Basic software

- install from Ubuntu Software store
  - Emacs, OnlyOffice, Chromium, VisualStudio Code
- install some basic packages with apt (`sudo apt install`) or GUI package manager
  - `gcc`, `make`, `perl`, `linux-headers-generic` (→ needed for guest add-ons)
  - `gnome-tweaks` (→ e.g. for swapping Caps and Ctrl keys)
  - `subversion`, `git`, `mercurial`, `openssh-server`
  - `recode`, `dos2unix`, `locales-all`, `jq`, `libxml2-dev`, `libxml2-utils`
  - Java: `default-jre`
- optional: install `Hoover` script (http://stefan-evert.de/Software.html#Hoover)
- install VirtualBox guest additions
  - insert CD from Devices menu, then allow software to run automatically + enter `issale` password
  - `sudo usermod -aG vboxsf issale`
    (→ so ISSALE user can access auto-mounted shared folders)
- give ISSALE user ownership of `/usr/local` to simplify software installation
  - `sudo chown -R issale /usr/local`
  - *note:* not recommended for regular Ubuntu installation with multiple users, perhaps better to make group-writable by `admin` group and add ISSALE user

Further software packages

- install R according to instructions at https://cran.r-project.org/bin/linux/ubuntu/
  - `sudo apt-add-repository 'deb https://cloud.r-project.org/bin/linux/ubuntu bionic-cran35/'`
  - `sudo apt-key adv --keyserver keyserver.ubuntu.com --recv-keys E298A3A825C0D65DFD57CBB651716619E084DAB9`
  - `sudo apt install r-base-dev`
  - `sudo apt install libopenblas-dev`
- install RStudio according to https://www.rstudio.com/products/rstudio/download/

    - simply download the `.deb` file and open with Software Install (default) in Firefox
- install Apache2 Web server and MySQL
    - `sudo apt install apache2 libapache2-mod-php`
    - `sudo a2enmod cgi`
    - `sudo apache2ctl restart`
    - `sudo apt install mysql-server mysql-client php-mysql`
    - *note:* need `sudo mysql` to login as MySQL root user and set up new accounts (because the MySQL package uses Unix authorization without password)
    - now configure Firefox and Chromium to show http://localhost/ as homepage (and default start-up page)
- basic Web server configuration
    - `sudo chown -R issale /var/www /usr/lib/cgi-bin`
    - *note:* not recommended for a regular multi-user linux installation; make writable for `admin` or `www-data` group and add ISSALE user to group
    - install Gopher CSS framework (http://www.stefan-evert.de/GOPHER/) into `/var/www/html/gopher`
    - create start page `/var/www/html/index.html` with links to all local pages and Web interfaces

Corpus indexing software

- install IMS Corpus Workbench (CWB) and Perl/CWB interface from http://cwb.sourceforge.net/ (Developers | SVN access)
    - check out source code from SourceForge SVN in working directory of your choice
    - `svn co http://svn.code.sf.net/p/cwb/code/cwb/trunk cwb`
    - `svn co http://svn.code.sf.net/p/cwb/code/perl/trunk cwb-perl`
    - install prerequisites for compilation
    - `sudo apt install autoconf bison flex gcc make pkg-config libc6-dev libncurses5-dev libpcre3-dev libglib2.0-dev libreadline-dev`
    - compile and install CWB command-line tools and CQP query processor
    - `cd cwb/`
    - `make clean all install realclean PLATFORM=linux-64 SITE=standard`
    - install Perl interface and command-line utilities
    - `cd ../cwb-perl/` → then in each subdirectory do
        - `perl Makefile.PL; make all test install clean`
    - create corpus data directories and install DICKENS demo corpus
        - `mkdir -p /usr/local/share/cwb/{registry,data}`
        - obtain DICKENS (and optionally EUROPARL) from http://cwb.sourceforge.net/download.php#corpora
        - install index files (`data/*`) under `/usr/local/share/cwb/data/Dickens`
        - copy registry file to `/usr/local/share/cwb/registry`
        - then adjust file paths in registry file
        - `cwb-regedit DICKENS :home /usr/local/share/cwb/data/Dickens :ifile /usr/local/share/cwb/data/Dickens/.info`

        - check that corpus has been installed properly
        - `cwb-describe-corpus -s DICKENS`
- optional: install CQPDemo (Dickens) Web interface in `/var/www/html/CQP/Dickens` and `/usr/lib/cgi-bin/CQP/Dickens`
    - need to adapt URLs in HTML files and scripts accordingly
    - *note:* CQPDemo source code is not available for download, but you can copy over these directories from the ISSALE VM
- optional: install Europarl GUI from SVN repository
    - `svn checkout http://svn.code.sf.net/p/cwb/code/gui/europarl/trunk/html /var/www/html/CQP/Europarl`
    - `svn checkout http://svn.code.sf.net/p/cwb/code/gui/europarl/trunk/cgi-bin /usr/lib/cgi-bin/CQP/Europarl`
    - adapt URLs in CGI scripts if necessary
    - must also download Europarl 3 corpus from http://cwb.sourceforge.net/download.php#corpora and install all 6 language components similar to procedure for Dickens above
- install UCS toolkit for co-occurrence data from http://www.collocations.de/software.html
    - `cd /usr/local/share`
    - `svn checkout svn://svn.code.sf.net/p/multiword/code/software/UCS/trunk UCS`
    - `sudo apt install libterm-readkey-perl libterm-readline-gnu-perl libtk-pod-perl libexpect-perl a2ps`
    - now configure the UCS installation
    - `(cd UCS/System; perl Install.perl)`
    - clean up backup files and link the command-line tools into search path
    - `Hoover -vr UCS` (if you didn't install Hoover: `rm UCS/System/bin/*~`)
    - `cd /usr/local/bin; ln -s ../share/UCS/System/bin/ucs* .`

CQPweb

- install CQPweb following the procedure in the CQPweb Admin Manual (http://cwb.sourceforge.net/files/CQPwebAdminManual.pdf)
    - `cd /var/www/html`
    - `svn checkout http://svn.code.sf.net/p/cwb/code/gui/cqpweb/trunk cqpweb`
        - Web server needs write access to CQPweb directory tree
        - `sudo chgrp -R www-data cqpweb`
        - `sudo chmod -R g+rwX cqpweb`
    - edit PHP configuration: `vscode /etc/php/7.2/apache2/php.ini`
        - change the following settings (use search to locate lines)
        - `memory_limit = 512M`
        - `max_execution_time = 600`
        - `upload_max_filesize = 128M`
        - `post_max_size = 128M`
        - `mysqli.allow_local_infile = On` (default changed in 04/2019)
        - enable (= uncomment) extensions: `mysqli`, `gd2`
        - save write-protected file with Retry as sudo

- create data directories for CQPweb and give Web server write permissions
  - `mkdir -p /usr/local/share/cqpweb/{data,registry,cache,upload}`
  - `sudo chgrp www-data /usr/local/share/cqpweb/*`
  - `sudo chmod g+rwx,o-rwx,+s /usr/local/share/cqpweb/*`
- configure Apache2 for CQPweb
  - create file `/etc/apache2/sites-available/cqpweb.conf` with the following content

```
<Directory "/var/www/html/cqpweb/">
   AllowOverride All
   Require all granted
</Directory>
```

  - `sudo a2ensite cqpweb`
  - `sudo apache2ctl restart`
- create MySQL user account and database for CQPweb
  - `sudo mysql -u root` then enter the following SQL commands
    - `UNINSTALL PLUGIN validate_password;` (→ avoid complaints about our weak password)
    - `create database cqpweb default charset utf8;`
    - `create user cqpweb identified by 'issale';`
    - `grant all on cqpweb.* to cqpweb;`
    - `grant file on *.* to cqpweb;`
    - `exit;`
- complete the CQPweb configuration
  - `cd /var/www/html/cqpweb/bin`
  - `php autoconfig.php`
    - enter admin user: `issale`
    - specify CQPweb directories created above, i.e. `/usr/local/share/cqpweb/data`, …
    - specify database configuration as specified above, i.e. account `cqpweb`, password `issale`, database `cqpweb`
  - `php autosetup.php`
    - admin user password: `issale`
- test CQPweb by installing pre-indexed Dickens corpus
  - copy registry file to CQPweb registry `cp /usr/local/share/cwb/registry/dickens /usr/local/share/cqpweb/registry`
  - in CQPweb Admin Control Panel, select Install Corpus and then click on … already indexed in CWB
  - enter corpus name `dickens` (lowercase!) and go through the usual configuration and indexing steps (see CQPweb Admin Manual)
  - *note:* in order to make the novel titles searchable by CQPweb, you need to create and upload an external metadata table (or Create minimalist metadata table)
  - should also upload and activate the `Arial-small.css` style sheet (smaller fonts with proper kwic display) and standard simple tagset definitions for CEQL queries (*note:* not available for download)
  - in Users and privileges | Manage privileges menu, create default access privileges for frequency lists + standard access to public

corpora (initialized with `dickens`) → grant these privileges to group `everybody`

BootCaT

- not compatible with current Java versions, so must install Java 8:
  `sudo apt install openjdk-8-jre`
- unpack ZIP archive as folder `/usr/local/share/bootcat`
- create shell script wrapper bootcat in this directory with content:

```
#!/bin/sh
JAVA=/usr/lib/jvm/java-8-openjdk-amd64/bin/java
$JAVA -jar /usr/local/share/bootcat/bootcat_frontend.jar
```

- and link it into search path:
  `chmod 755 /usr/local/share/bootcat/bootcat`
  `ln -s /usr/local/share/bootcat/bootcat /usr/local/bin`
- if you want a clickable icon on your desktop, create a file `~/Desktop/BootCaT.desktop` with the following content:

```
[Desktop Entry]
Encoding=UTF-8
Name=BootCaT
Comment=Launch BootCaT GUI
Exec=/usr/local/bin/bootcat
Icon=/usr/local/share/bootcat/bootcat_frontend.ico
Type=Application
Terminal=false
```

- you can also put this file into `/usr/share/applications` to make it available for all users in the application launcher

Python packages

- corpus linguists and NLP researchers should only use Python 3
  - `sudo apt install ipython3 jupyter`
- install the following packages with package manager or `sudo apt install`
  - `csvkit`, `csvkit-doc` (→ command-line tools for manipulating CSV files)
  - *note:* you can read the documentation with `xdg-open /usr/share/doc/csvkit/html/index.html` (and accordingly for other `-doc` packages)
  - `python3-pip`, `virtualenv` (→ for installing Python packages)
  - `python3-scrapy`, `python-scrapy-doc` (→ Web scraping framework)
  - `python3-numpy`, `python3-scipy`, `python3-pandas`, `python3-sklearn` (→ data science stack required by many NLP tools)
  - `python-numpy-doc`, `python-scipy-doc`, `python-pandas-doc`, `python-sklearn-doc` (→ corresponding documentation)
  - `python3-regex` (→ PCRE-style regular expressions)
  - `python3-nltk` (→ NLTK toolkit for basic NLP tasks)
  - `python3-tweepy`, `python-tweepy-doc` (→ easy Twitter API)
  - to be continued …

- install additional packages from PyPI sources, using the pip package manager
  - in our Ubuntu setup, always use `pip3 install --system` to install packages in Python 3 for all users
  - *note:* on regular Linux installation, may need `sudo pip3 install --system` (if user account doesn't have write permissions in `/usr/local` tree)
  - *note:* if Anaconda Python, simply use `pip` (*not* `pip3`) without further options
- install interfaces to CWB and the CQP query processor
  - `pip3 install --system cwb-python`
  - the CQP interface in the current cwb-python distribution is broken, so we need to overwrite it with a patched version from a temporary package: `pip3 install --system http://www.collocations.de/temp/PyCQP_interface-1.0.1.tar.gz`
  - see course slides for an example of accessing CWB and CQP from Python
- might also install binaries of `gab_lemmatizer` and `add_glemma`
  - non-public software, copy binaries from some other Linux server
  - adjust paths in the scripts and link into `/usr/local/bin` w/o extension

Activation of the Ubuntu ISSALE VM by end users

- when installation and configuration is complete, export the VM in `.ova` format via menu File | Export Appliance
- end users need to install VirtualBox, then select menu File | Import Appliance
  - select the `.ova` file and accept default setting
  - reminder: all passwords in the VM are `issale`
- create a shared folder (say `Ubuntu ISSALE/`) somewhere on the host computer for easy file exchange with the VM
  - safer than giving VM access to entire file system or user home
  - if you always work in shared folder within VM, you can install a new version of the VM without losing data (but you will also be able to update packages and installed software directly in the VM)
- set configuration options while VM is still turned off (click Settings icon in VBox)
  - General | Advanced | Shared Clipboard = Bidirectional
  - System | Motherboard → at least 4 GB RAM (more if you can afford it)
  - System | Processor → enable multiple CPU cores (if you can afford it)
  - Display | Screen → at least 64 MB video RAM, enable Acceleration if possible
  - Shared Folders → add your shared folder (auto mount, *not* read only) and specify folder name `issale`
  - Network | Adapter 1 ➞ enabled, Attached to = NAT, then
  - Advance | Port Forwarding ➞ create entries
    - 127.0.0.1 port 8080 (host) to 10.0.2.15 port 80 (guest)
    - 127.0.0.1 port 2222 (host) to 10.0.2.15 port 22 (guest)
- Ubuntu guest configuration
  - Settings | Region & Language | Input Source → add your keyboard layout (keyboard switcher & viewer available from menu bar at top of screen)
- integration with host computer
  - shared folder will be mounted as `/media/sf_issale` in guest
  - access VM Web server from host: http://localhost:8080/
  - SSH from host: `ssh -p 2222 issale@localhost`