

---

# Elpis

*Release 1.0.6*

**Ben Foley, Nicholas Lambourne, Nay San, Matthew Low, Harry Ke**

**Sep 05, 2023**



## INSTALLING ELPIS

<b>1</b>	<b>Installing Elpis on your computer</b>	<b>3</b>
<b>2</b>	<b>Install Elpis on Google Cloud for Kaldi</b>	<b>5</b>
<b>3</b>	<b>Install Elpis on Google Cloud with GPU</b>	<b>7</b>
<b>4</b>	<b>Getting started</b>	<b>11</b>
<b>5</b>	<b>Overview</b>	<b>13</b>
<b>6</b>	<b>Setup</b>	<b>15</b>
<b>7</b>	<b>Transcription types</b>	<b>17</b>
<b>8</b>	<b>About the steps</b>	<b>19</b>
<b>9</b>	<b>Recordings</b>	<b>21</b>
<b>10</b>	<b>Pronunciation Dictionary</b>	<b>25</b>
<b>11</b>	<b>Training sessions</b>	<b>29</b>
<b>12</b>	<b>Making a new transcription</b>	<b>33</b>
<b>13</b>	<b>More information about training files</b>	<b>35</b>
<b>14</b>	<b>Preparing files</b>	<b>37</b>
<b>15</b>	<b>Viewing Elpis training log file</b>	<b>41</b>
<b>16</b>	<b>How to setup repositories to develop Elpis</b>	<b>43</b>
<b>17</b>	<b>Building the Elpis Docker image</b>	<b>45</b>
<b>18</b>	<b>Making a release</b>	<b>47</b>
<b>19</b>	<b>Using the CLI Elpis Python API</b>	<b>49</b>
<b>20</b>	<b>Using the CLI for Elpis GPU</b>	<b>51</b>
<b>21</b>	<b>Handy GCP commands</b>	<b>53</b>



Elpis is a tool which language workers with minimal computational experience can use to build their own speech recognition system for automatically transcribing audio.



## INSTALLING ELPIS ON YOUR COMPUTER

Elpis can be installed with Docker, a virtual computer running on **your** computer. To use this version of Elpis, you first need to install Docker.

Docker is a program which helps standardise the way we do computational tasks with data, regardless of the operating systems of all the people who might want to run those tasks. Rather than building separate code for Windows, Linux, Mac operating systems, we can write once and run it on a myriad of operating systems using Docker. For more information about Docker, view [Nay San's slides](#).

Follow the instructions on the [Docker site](#) to install Docker. You will need to create a (free) account with them to be able to download the Docker installer.

After you have installed Docker, start it. On a Mac, you will see a little whale icon in the top menu bar. On Windows you'll see a whale icon in the system tray.

With Docker running, we will use a **terminal** to install Elpis.

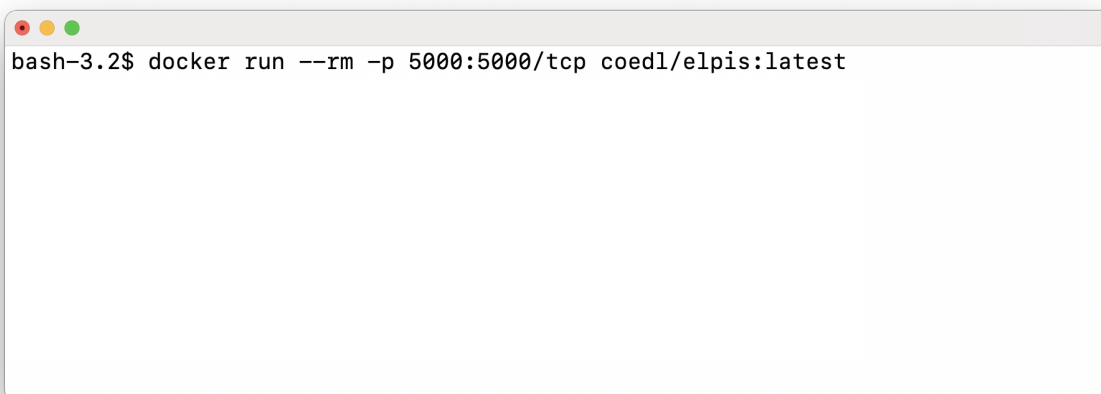
When you use an application like Elan or Word, you are using a 'graphical user interface (GUI)' to do stuff to your data via menus and buttons. Another way of interacting with your computer is via a terminal, also known as a command line or command prompt.

On Mac, open the *Terminal* app in your *Applications > Utilities* folder.

For Windows, open the search field in your taskbar, type `command` or `cmd` into it. Then, click or tap on the *Command Prompt* result to open it.

Download and run the Elpis Docker image by pasting this command in a terminal and pressing Return (or Enter).

```
docker run --rm --name elpis -p 5001:5001/tcp -p 6006:6006/tcp coedl/elpis:latest
```



run command

If this is the first time you have run the command, you should see a message “Unable to find image ‘coedl/elpis:latest’ locally”. All this means is that Docker has looked to see if there’s a local copy of the Docker image, and couldn’t find one. It will then start to download the image in a series of “layers”. Each layer will go through a process of Waiting and Pulling (pulling involves Downloading and Extracting). When all layers are complete, Docker will create a container from the image and start Elpis in the container.

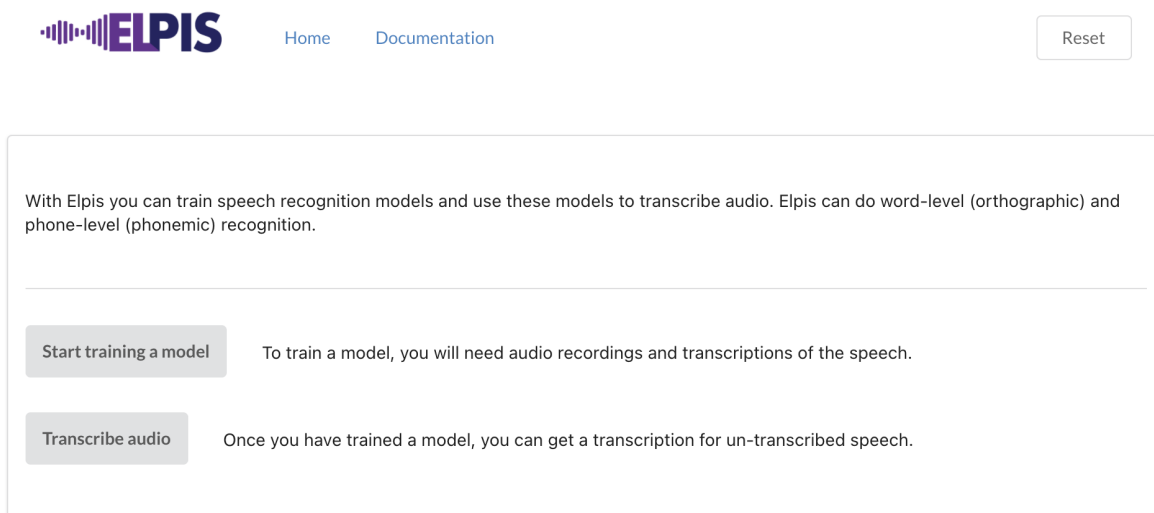
When you see a message about the server running, open <http://0.0.0.0:5001> in a browser. If you are on a Windows machine, try <http://localhost:5001> instead.

```
bash-3.2$ docker run --rm -p 5000:5000/tcp coedl/elpis:latest
Skipping virtualenv creation, as specified in config file.
* Serving Flask app "elpis" (lazy loading)
* Environment: development
* Debug mode: on
* Running on http://0.0.0.0:5000/ (Press CTRL+C to quit)
* Restarting with stat
* Debugger is active!
* Debugger PIN: 276-574-261
```

Docker

running

You should see the Elpis interface. It might look a little different to this, depending on changes in the current version.



Docker

welcome screen

With Elpis going, follow the steps in the [Elpis workshop guide](#).



## INSTALL ELPIS ON GOOGLE CLOUD FOR KALDI

If this is your first time using Elpis on Google Cloud, follow the steps on the [Setup Google Cloud account](#) page.

When you have finished using Elpis on a GCP virtual machine, make sure you stop it to prevent ongoing costs.

### 2.1 Create a Virtual Machine

Go to the [Compute Engine > VM Instances](#) page.

Click “Create”

- Name it
- Select a Region and Zone
- Choose a machine size. The size will determine how much it will cost to run. The smallest vCPU instance on GCP is more than enough for the toy data set. That’s first series N1 with g1-small (shared-core machine type with 0.5 vCPU, 1.70 GB of memory, backed by a shared physical core). At training time CPU usage peaks at about 10% of the available compute on that instance type. For a bigger data set, or for faster training & transcription time, choose a faster machine.
- Increase the boot disk size to 20GB plus the size of your training corpus. For example, to train with 5GB of audio and ELAN files, set the disk size to 25GB.
- Select HTTP and HTTPS traffic in the Firewall options
- Click the “Management, security, disks, networking, sole tenancy” link.

Paste the following code into the “Startup Script” box

```
sudo apt-get update
sudo apt-get install -y ca-certificates curl gnupg lsb-release
sudo mkdir -p /etc/apt/keyrings
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /etc/apt/
↪keyrings/docker.gpg
echo "deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.gpg] \
https://download.docker.com/linux/ubuntu $(lsb_release -cs) stable" | \sudo tee /etc/apt/
↪sources.list.d/docker.list > /dev/null
sudo apt-get update
sudo apt-get install -y docker-ce
sudo chmod 666 /var/run/docker.sock
mkdir /state
sudo docker run -d --name elpis -v /state:/state -p 80:5001/tcp coedl/elpis:latest
```

Then press “Create”

It will take between 15 and 30 minutes for the machine to start up and install all the software.

## 2.2 For multiple machines

Make an instance template with the same settings.

Install `gcloud`

Create multiple machines with this command. Replace the zone and template values. Size is the number of instances.

```
gcloud compute instance-groups managed create elpis-group \  
  --zone "us-central1-a" \  
  --template "elpis-medium-template" \  
  --size 2
```

## INSTALL ELPIS ON GOOGLE CLOUD WITH GPU

If this is your first time using Elpis on Google Cloud, follow the steps on the [Setup Google Cloud account](#) page.

This document will go through a process of enabling the network access required to view training progress with Tensorboard, and detail the steps to start a machine running Elpis.

When you have finished using Elpis on a GCP virtual machine, make sure you stop it to prevent ongoing costs.

### 3.1 Enable network access

Elpis uses Tensorboard to display training progress and plots. To enable us to view the Tensorboard page, we need to add a “firewall rule” in the Cloud console.

Sign in to the console. If you have multiple projects, choose the one you want to work with.

In the left hand navigation menu, go to “VPC network > Firewall”. Click “Create Firewall Rule” (blue button at the top of the page).

Use the following settings, then click Create. This will create a rule which our machine can use to enable browser traffic to reach the Tensorboard.

- Name: tensorboard
- Direction of traffic: Ingress
- Target tags (make sure this is lowercase, and all one word): tensorboard
- Source IPv4 ranges: 0.0.0.0/0
- Protocols and ports: Specified protocols and ports
- TCP: 6006

### 3.2 Create a Virtual Machine and run Elpis

Go to the `Compute Engine > VM instances` page.

To run Elpis, create an instance with the following settings. These resources will be adequate for a small amount of data, but may need to be increased depending on the quantity of your data. This configuration would cost approximately \$600 to run all day, every day, for a month.

- Name: Give your instance a meaningful name, perhaps the name of the language you are training with.
- Region and zone: These can be left as is, or change to a location near you if required. Note that different regions may have different GPU options.

- Machine family: GPU
- GPU-type: NVIDIA T4
- Number of GPUs: 1
- Machine-type: n1-standard-16 (16 vCPUs, 60 GB memory)

Scroll down to the Boot disk section. Change the boot disk to use the following settings.

- Operating system: Ubuntu
- Version: Ubuntu 20.04 LTS x86/64
- Boot disk type: Standard persistent disk
- Size (GB): 300

Scroll down to the “Firewall” settings. Tick Allow `http` traffic

Click “Advanced options” to open that section.

Click “Networking” to open that section.

Type `tensorboard` in the Network tags field. This will allow the virtual machine to use the Tensorboard firewall rule we created earlier.

Scroll down and click on Management, and paste the following code into the Automation Startup script section. This code will install all the required software, download Elpis to the VM, and start Elpis.

Note that we install this way, and not using image deploy because image deploy limits the OS to “container optimised”, which prevents use of `--gpus all` docker run flag. To use `--gpus all` flag, we need to install specific version of nvidia drivers, not container optimised.

```
# GPU startup script v0.6.4

# Check if this has been done before. Skip driver installation if so, just run Elpis
if [[ -f /etc/startup_installed ]];
then
    sudo chmod 666 /var/run/docker.sock
    # Run Elpis (non-interactive so that Elpis starts automatically)
    docker run -d --rm --name elpis --gpus all -v /state:/state -p 80:5001/tcp -p
    ↪6006:6006/tcp coedl/elpis:latest
    exit 0;
fi

# Otherwise, install all the things.. then run Elpis
# Install CUDA

sudo apt install linux-headers-$(uname -r)
curl -O https://developer.download.nvidia.com/compute/cuda/repos/ubuntu2004/x86_64/cuda-
↪ubuntu2004.pin
sudo mv cuda-ubuntu2004.pin /etc/apt/preferences.d/cuda-repository-pin-600
wget https://developer.download.nvidia.com/compute/cuda/repos/ubuntu2004/x86_64/cuda-
↪keyring_1.0-1_all.deb
sudo dpkg -i cuda-keyring_1.0-1_all.deb
sudo add-apt-repository "deb https://developer.download.nvidia.com/compute/cuda/repos/
↪ubuntu2004/x86_64/ /"
sudo apt update
sudo apt -y install cuda
```

(continues on next page)

(continued from previous page)

```

# Install NVIDIA Container Toolkit

curl https://get.docker.com | sh \
  && sudo systemctl --now enable docker

distribution=$(. /etc/os-release;echo $ID$VERSION_ID) \
  && curl -s -L https://nvidia.github.io/nvidia-docker/gpgkey | sudo apt-key add - \
  && curl -s -L https://nvidia.github.io/nvidia-docker/$distribution/nvidia-docker.list_
  ↪ | sudo tee /etc/apt/sources.list.d/nvidia-docker.list

sudo apt-get update
sudo apt-get install -y nvidia-docker2
sudo systemctl restart docker
sudo usermod -aG docker $USER
sudo chown $USER /var/run/docker.sock
sudo chmod 666 /var/run/docker.sock

# Handy little app to check NVIDIA GPUs stats
sudo apt -y install nvidia-smi

# Get elpis
cd ~
git clone https://github.com/CoEDL/elpis.git

# Will make it easier to copy model files etc out of the container
mkdir state

# Make a file which can be detected on next startup and thus skip doing this every time
touch /etc/startup_installed

echo "done"

# Download and run Elpis (non-interactive so that Elpis starts automatically)
docker run -d --rm --name elpis --gpus all -v /state:/state -p 80:5001/tcp -p 6006:6006/
  ↪ tcp coedl/elpis:latest

```

This startup script will only run the first time the VM starts, to reduce the instance load time on subsequent restarts.

Then, scroll to the bottom of the page and click “Create”. The page will redirect to the virtual machine list, and show the status of the machine starting up.

After the machine starts, it can take up to 15 minutes for everything in the startup script to be installed. Wait 15 minutes or so, and then copy the External IP address.

Open a browser. In the browser’s location field, type `http://` and paste the IP address. It should end up looking like `http://34.125.96.234`. Then press enter/return to go to your Elpis machine.

With Elpis going, follow the steps in the [Elpis workshop guide](#).

### 3.3 Adding projects (optional)

Later, you may wish to add a new project to separate the usage of services across different experiments or activities.

Click the project list in the top blue menu. In the popup, click “New Project”.

On the New Project screen, add a project name and press “Create”.

When the project has been created, you will be prompted to select it. Having done that, the page will show the project’s Dashboard.

**GETTING STARTED**





## **OVERVIEW**

The speech recognition process (also called speech to text) broadly involves steps of:

- Organising files that will be used to train the system
- Making pronunciation rules for your language
- Acoustic, pronunciation and language training

Then, using the trained system we can get a new transcription on un-transcribed recordings.



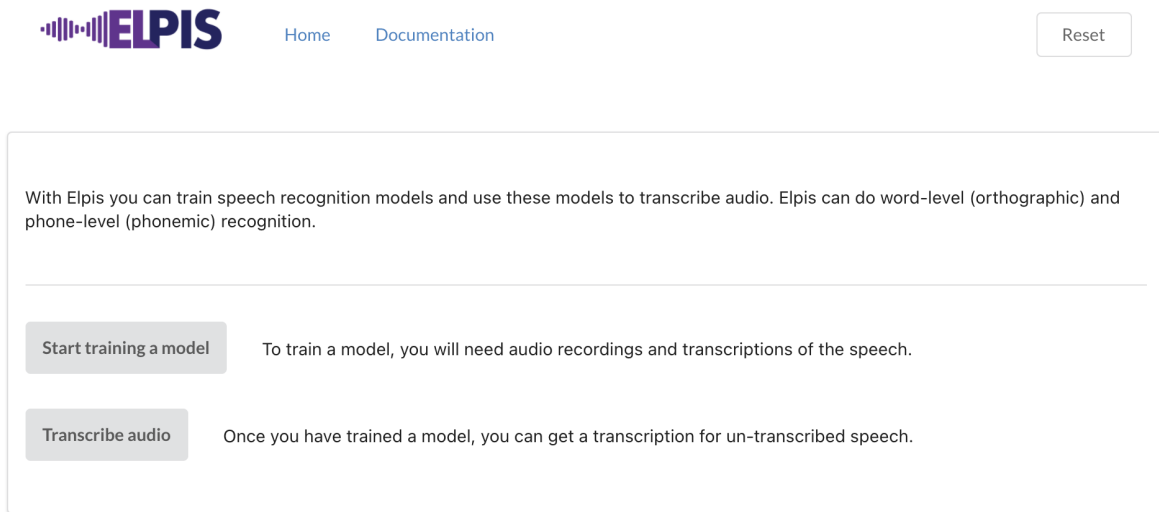
## 6.1 Get some training files

Start with downloading some files to use during the workshop. [Download a zip of the files here](#). The recordings in this zip are Abui language (abz) provided by František Kratochvíl, and Yongning Na language (nru) provided by Alexis Michaud and Oliver Adams.

After the zip file has downloaded, unzip it to create a folder somewhere handy (for example, your Desktop).

## 6.2 Start Elpis

- We will provide a list of servers on the workshop day.
- Get an address from the list.
- If you are using Elpis in Docker on your own computer, the address will be <http://0.0.0.0:5001> (or, if that doesn't work try <http://localhost:5001>).
- Open a new web browser (Chrome or Firefox).
- Paste the address into the location bar.
- Press Enter/Return to start Elpis.
- When Elpis starts it looks like this.



This is the Welcome page. To come back here anytime, you can click **Home** in the top menu.

Also in the top menu is a link to this Elpis documentation.

The **Reset** button will reset Elpis, removing all the recordings and models that you have started. It won't affect any of your original files.

On the Welcome page we have two options: to *train a model* and to *transcribe audio*. First, we need to train the speech recognition system. We are planning to include some pre-trained systems in Elpis to make things easier. For now, we will start by training Elpis, so click **Start training a model**.

Once we have trained a system we can use it to transcribe.

*A model is the system that Elpis learns about the language, based on the recordings and transcriptions that you provide.*

---

## TRANSCRIPTION TYPES

Elpis has two transcription methods. It can be trained to recognise *words* or *phonemes*.

To train Elpis to recognise words in speech, we will need to provide

1. some recordings
2. transcriptions of the audio
3. and for word recognition it needs some information about the way the words are pronounced.

For word recognition, Elpis will try to create a pronunciation dictionary from some rules which we give it in a [letter-to-sound](#) file. We'll see an example of this soon. In other tools this is called grapheme-to-phoneme or G2P.

The Elpis phoneme level recognition method doesn't require the pronunciation rules, just audio and transcriptions.

Elpis currently uses ELAN format for the transcriptions. We are working on supporting other formats, please let us know what you need. Transcriptions don't need to be at individual word or phoneme level, the best length is at "utterance" level of around ten second duration. For more information about preparing your own files, see the [Preparing files](#) page.

For this workshop, we will choose the **Word** method.

[Home](#)[Documentation](#)[Reset](#)

### Choose a type

Word

This will require audio files, transcriptions and a 'letter to sound' file.

Phoneme

Requires audio and transcriptions.



## ABOUT THE STEPS

There are four main steps in Elpis, with sub-steps in each.

1. Recordings
2. Pronunciation Dictionary
3. Training
4. New transcriptions

**Recordings** is where we collect and prepare the audio and text to train Elpis.

The **Pronunciation Dictionary** is where the system works out how the text words from the Recordings step are pronounced. This step is skipped when doing phoneme transcription.

**Training** is where the speech recognition models are built.

**New transcriptions** is the place we go to use an existing training session to obtain a first-pass transcription on new audio.

---





## RECORDINGS

We can do multiple sessions with Elpis. To keep track of which group of files we are using, give them a name here. For example, if you are using the Abui sample recordings, you could name this “Abui recordings”. Then click Add New

The screenshot shows the Elpis web interface. At the top left is the Elpis logo (a purple waveform icon followed by the text 'ELPIS'). To its right are links for 'Home' and 'Documentation'. On the top right, there is a green circular button labeled 'word' and a 'Reset' button. The main content area is titled 'Recordings'. On the left side of this area is a sidebar menu with the following items: 'Recordings' (highlighted in purple), 'Files', 'Wordlist', 'Pronunciation' (with sub-items 'Letter to sound' and 'Dictionary'), 'Training' (with sub-items 'Settings', 'Train', and 'Results'), and 'New transcriptions'. The main content area under the 'Recordings' header contains a purple box with the text 'Make a new one using the form below.' Below this is a form with a 'Name' label and a text input field containing 'Abui recordings'. At the bottom of the form is a grey button labeled 'Add New'.

### 9.1 Add files

On the *Add files* page, click inside the dotted area and go to where you downloaded the Abui files. Open the transcribed folder, select all the *wav* and *eaf* files and add them.

You can add additional words by uploading a wordlist in a plain text file named `additional_word_list.txt`, or a text corpus (with sentences) named `corpus.txt`. These are optional files. Words in either of these uploaded files will extend the pronunciation lexicon. Content in `corpus.txt` will also be used by the language model.

For more information about preparing your own files for Elpis, see the [wiki page](#).

**Recordings**

**Files**

Wordlist

**Pronunciation**

Letter to sound

Dictionary

**Training**

Settings

Train

Results

**New transcriptions**

## Add files

Current type: word  
Current recordings: Abui recordings

Upload your language recording and transcription files here.  
Please use 44.1kHz WAV format audio, and .eaf Elan transcription files.  
Audio files and transcription files need to have matching filenames.  
You can also upload text files like wordlists or stories that don't have audio.

Upload or drop your files here

Upload

Next

### Import Settings

No settings.

Next

## 9.2 Select tiers

Elan files can have multiple tiers for transcription, glosses, translations, etc. For training, we need to select the tier that contains the transcription text.

Elpis reads the Elan files you uploaded. The tier names and tier types from the files are shown here to choose from, or you can choose a tier by order - the top-most tier in all files would be selected by choosing 0, the second tier would be selected by choosing 1.

Select one of the Tier options. For the Abui files, choose **Tier Name** for the Selection, and **Phraseas** the Tier name.

For this workshop there is no need to change the punctuation settings. For more info about what this setting does please get in touch.

Then click **Next**.



Recordings

Files

Wordlist

Pronunciation

Letter to sound

Dictionary

Training

Settings

Train

Results

New transcriptions

## Add files

Current type: word

Current recordings: Abui recordings

Upload your language recording and transcription files here.  
Please use 44.1kHz WAV format audio, and .eaf Elan transcription files.  
Audio files and transcription files need to have matching filenames.  
You can also upload text files like wordlists or stories that don't have audio.

Upload or drop your files here

Upload

### Uploaded files

Audio files	Transcription files
abui_1.wav	abui_1.eaf
abui_2.wav	abui_2.eaf

Next

### Import Settings

Tiers

Choose the tier that your transcriptions are on, just choose one of these.

Selection Mechanism

tier\_name

Tier name

Phrase

Punctuation

What to do with punctuation.

Replace these with spaces

!"#\$%&'()\*+,-./:;<=>@[\\

Remove these

Cleaning

Remove text from annotations, add one per line.

Words to remove

<silence>


Tags to remove

@eng@

Next

## 9.3 Prepare

On the *Prepare* page we can see how Elpis has read your transcription files. If you have lots of training text there will be a delay while the text is prepared.

[Home](#) [Documentation](#)

word Reset

**Recordings**

Files

**Wordlist**

**Pronunciation**

Letter to sound

Dictionary

**Training**

Settings

Train

Results

**New transcriptions**

### Wordlist

Current type: word  
Current recordings: Abui recordings

#### Word frequency of training files


Next

Word	Frequency
amakaang	1
di	1
kaai	1
hada	1
muila	2
dining	1
ayoku	1
kamar	1
mia	1
mui	1

## PRONUNCIATION DICTIONARY

For a word recognition system, the pronunciation dictionary is made so the system knows how words are pronounced. Elpis will make a rough draft for the words in the wordlist, based on a letter-to-sound file which you provide. This step is not required for phoneme recognition.

Like the recordings step, give this step a name. For example “Abui pronunciation”

[Home](#) [Documentation](#)

word

Reset

**Recordings**  
Files  
Wordlist

**Pronunciation**  
Letter to sound  
Dictionary

**Training**  
Settings  
Train  
Results

**New transcriptions**

### Pronunciation Dictionary

Make a new one using the form below.

Name

Abui pronunciation

Select a group of recordings

Abui recordings

Add New

## 10.1 Letter to sound rules

The letter-to-sound file is a text file of rules mapping your orthography into phonemic transcription. Elpis will use it to build a pronunciation dictionary for the words in the transcriptions you uploaded.

It is formatted in two columns, space separated. The left column is all the characters in your corpus. The right column is a single symbol representing the sound.

Comments can be written in the file with a # starting the comment line.

Here's a section of the Abui one:

```
# Abui
j J
f f
s s
h h
m m
n n
ng ŋ
r r
```

Note that the file has to have particular text format. On Windows, use the free utility Notepad++ to convert CrLf to Lf in one go (Edit > EOL Conversion > Unix Format).

Upload the letter to sound rules `letter_to_sound.txt` from the Abui folder.


[Home](#)
[Documentation](#)

word

Reset

Recordings

Files

Wordlist

Pronunciation

Letter to sound

Dictionary

Training

Settings

Train

Results

New transcriptions

## Upload Letter to Sound rules

Current type: word

Current pronunciation dictionary: Abui pronunciation

Current recordings: Abui recordings

The letter to sound file is used to build a pronunciation dictionary for the corpus. Make one by listing one column of all the characters in your corpus. Make a second column (separated by a space) of a symbol representing how that character is pronounced. You can use IPA, or SAMPA for the pronunciation symbols. You can include comments in this file by beginning the comment line with #. For example:

```
# This is a comment
n n
ng ŋ
r r
y j
```

Next

```
# Consonants
p p
b b
t t
d d
k k
g g
ʔ ʔ
c C
j J
f f
s s
```

## 10.2 Pronunciation


Elpis uses the letter to sound file we uploaded to make a breakdown of how each word in our training files might be pronounced. For some languages the simple technique that Elpis uses will be accurate, for other languages, the results will need to be corrected.

Scroll through the list and review the results.

To make corrections, you can type your changes in this field. After making corrections, press Save.

If you need to undo your changes, press the **Reset** button that is below the pronunciation text to reset back to the rough draft.

If you notice characters in brackets e.g. (h), this indicates that the word includes a letter that is not covered in the letter-to-sound file. To correct this, add a letter to sound line in your letter-to-sound file for this letter, go back and make a new Pronunciation Dictionary, then upload the corrected letter-to-sound file again.


[Home](#)
[Documentation](#)

word

Reset

Recordings

Files

Wordlist

Pronunciation

Letter to sound

Dictionary

Training

Settings

Train

Results

New transcriptions

## Review the pronunciation dictionary

Current type: word

Current pronunciation dictionary: Abui pronunciation

Current recordings: Abui recordings

The pronunciation dictionary is generated from the wordlist and letter-to-sound rules you uploaded.

Characters surrounded by parenthesis were not found in your letter to sound map. Please add them, re-upload the l2s file and check the lexicon again.

Next

```

!SIL sil
<UNK> spn
hada h a d a
mia m I a
mui m u I
di d I
amakaang a m a k a: ɲ
muila m u I l a
kamar k a m a r
kaai k a: I
ayoku a j O k u
dining d I n I ɲ

```

Reset

Save

The !SIL and <unk> lines are used to handle silence and unknown words.

Check words that have been transcribed with consecutive matching characters. Do they represent one sound or two? If only one, add a line to your `letter-to-sound.txt` file, mapping the consecutive characters to a single symbol and rebuild the lexicon.

For example, if `wunne` is mapped to `wunne w n n` in the lexicon, then add `nn n` to `letter-to-sound.txt`, upload it again and rebuild the lexicon. The results should be collapsed lexicon entry `wunne w n`.

If your language has digraphs, put these earlier in the l2s, above single characters. For example,

```

ng ɲ
n n

```



## TRAINING SESSIONS

Now our training files have been prepared, we can start a new training session. Give it a name then click Next.

The screenshot shows the ELPIS web interface. At the top left is the ELPIS logo. To its right are links for 'Home' and 'Documentation'. On the top right, there is a green circular button labeled 'word' and a 'Reset' button. The main content area is titled 'Training' and contains a sidebar on the left with the following menu items: 'Recordings' (with sub-items 'Files' and 'Wordlist'), 'Pronunciation' (with sub-items 'Letter to sound' and 'Dictionary'), 'Training' (highlighted), 'Settings', 'Train', 'Results', and 'New transcriptions'. The main panel for 'Training' has a purple box with the text 'Make a new one using the form below.' Below this is a 'Name' input field containing 'Abui training'. Underneath is a dropdown menu labeled 'Select a pronunciation dictionary for this session' with the selected option 'Abui pronunciation | Abui recordings'. At the bottom of this panel is an 'Add New' button.

### 11.1 Settings


Here you can adjust settings which affect the tool's performance.

If you are using the Kaldi model, you can set the “n-gram” value. A unigram (1) value will train the model on each word. A trigram (3) value will train the model by words with their neighbours.

For HFT models, if you are trying Elpis out on your own computer, change the default settings to the following. These lower settings will reduce the amount of memory required for training.

- Number of epochs: 1
- Min duration: 1

- Max duration: 10
- Batch size: 1
- For testing purposes, select “Debug using a subset of the data” if you have a lot of data. This will use a small sample of your training data to try it out.



[Home](#)
[Documentation](#)

word

Reset

**Recordings**

Files

Wordlist

**Pronunciation**

Letter to sound

Dictionary

**Training**

Settings

Train

Results

New transcriptions

## Settings

Current type: word

Current training session: Abui training

Current pronunciation dictionary: Abui pronunciation

Current recordings: Abui recordings

These settings are used to adjust the training.

The n-gram adjusts whether the system learns single words or words and their neighbours. A value of 1 will set the system to learn single words. To make the system learn from words and their neighbouring words, set the n-gram to 2 or 3.

1

▼

Next

## 11.2 Training

Go to the **Training** page and press **Start training** to begin.

During training, we will see progress through the stages. The terms here are speech recognition jargon words. Understanding what they mean is not required to use Elpis. Depending on the duration of your training recordings, Elpis can take a long time to train. During these long training processes, seeing the terms here can at least indicate that the process is still going. As each stage completes, it will show a little tick.


[Home](#)
[Documentation](#)

word

Reset

Recordings

Files

Wordlist

Pronunciation

Letter to sound

Dictionary

Training

Settings

Train

Results

New transcriptions

## Train

Current type: word

Current training session: Abui training

Current pronunciation dictionary: Abui pronunciation

Current recordings: Abui recordings

Settings

n-gram 1

▸ setup ✓

▸ acousticPreparation ✓

▸ featureExtraction ✓

▸ dataPreparation ✓

▼ modelCreation ✓

▸ monophoneTraining ✓

▸ triphoneTraining ✓

```

arpa2fst --disambig-symbol=#0 --read-symbol-table=data/lang/words.txt
data/local/tmp/lm.arpa data/lang/G.fst
LOG (arpa2fst[5.3.118~1-6d32a]:Read():arpa-file-parser.cc:94) Reading \data\ section.
LOG (arpa2fst[5.3.118~1-6d32a]:Read():arpa-file-parser.cc:149) Reading \1-grams: section.
LOG (arpa2fst[5.3.118~1-6d32a]:RemoveRedundantStates():arpa-lm-compiler.cc:359)
Reduced num-states from 1 to 1

```

trained

Start training

Next


## 11.3 Results

When training is complete, go to the Results page to see the results for this training session. These results tell us how the training went, and help us to understand what happened in the training process. These numbers are **scored** by comparing the words in one of the original transcriptions against the computer's version.

The results are:

- WER - Word Error Rate
- Count - a word count of how many words were wrong compared with the total number of words in the sample
- DEL - words that were deleted (missed)
- INS - words that have been inserted (added)

- SUB - words that have been substituted (mistaken)


[Home](#)
[Documentation](#)

word

Reset

Recordings

Files

Wordlist

Pronunciation

Letter to sound

Dictionary

Training

Settings

Train

Results

New transcriptions

## Results

Current type: word

Current training session: Abui training

Current pronunciation dictionary: Abui pronunciation

Current recordings: Abui recordings

Results are obtained by testing the model on a sample of audio from the recordings. The error rate is calculated by adding up all the words that are wrongly skipped, added, or swapped compared to what was spoken. The number of errors is then dividing by the total number of words that were spoken.

<b>Word error rate (WER)</b>	<b>100.00%</b>
Number of incorrect words / total words	5 / 5
Deletions: spoken words missed in the transcription (DEL).	4
Insertions: extra words added to the text that weren't spoken (INS).	0
Substitutions: correct words swapped for incorrectly recognised words (SUB).	1

## MAKING A NEW TRANSCRIPTION

Now the training has completed, go to the **New Transcriptions** step.

Click **Upload**, navigate to the files you downloaded and select the `audio.wav` file from the Abui untranscribed folder. Then click **Transcribe**.

The screenshot shows the ELPIS web interface. At the top left is the ELPIS logo. To its right are links for 'Home' and 'Documentation'. On the far right, there is a green circle containing the word 'word' and a 'Reset' button. The main content area is titled 'Transcribe some audio'. Below the title is a yellow box containing the following text: 'Current type: word', 'Current training session: Abui training', 'Current pronunciation dictionary: Abui pronunciation', and 'Current recordings: Abui recordings'. Below this is a dashed box with the text 'Upload or drop an audio file here' and an 'Upload' button. Below the dashed box is a text input field containing 'Uploaded audio.wav'. Below the input field is a 'Transcribe' button. At the bottom of the form is a light grey box containing the text 'ready'.

Again, we see progress through the transcription stages, and more speech recognition jargon!

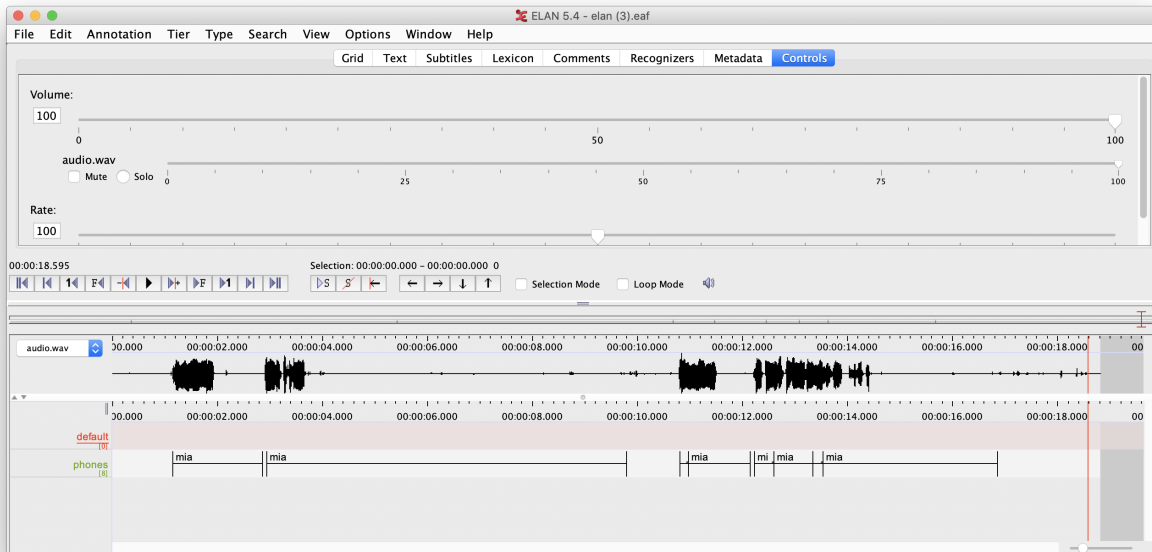
After the transcription is done, the transcription will show on the page, and the transcription can be downloaded in text or Elan format.

The transcription text may be shown in a range of tones from black to light grey. The darkness of the text is relative to how confident the system was about suggesting that text for that particular sound. This is known as “confidence”. When

you download the Elan file the confidence values are included on a tier. You can switch off the confidence display using the toggle switch on the right hand side of the panel.

[wiki/assets/latest/145-transcribed-confidence.png](https://wiki/assets/latest/145-transcribed-confidence.png)

Listen in Elan. You will need to move the audio file into the same location as the Elan file for Elan to link to it.



## MORE INFORMATION ABOUT TRAINING FILES

The system trains with existing audio recordings and transcriptions. Generally, the more hours of training recordings you can train with, the better the results. However, it's not simply a matter of throwing everything you have into a bucket. Time spent cleaning and fine-tuning your existing transcriptions will have a good impact on your results.

You will typically get better results with few hours of files by using recordings from a common recording activity, e.g. short sentences, or stories, or word-repetition exercises.

For Elpis, the file format requirements are:

- a) WAV audio, preferably 16kHz mono but the system can convert stereo files and resample from different sample rates.
- b) Orthographic transcription of the audio. For today's workshop, the interface is using Elan transcriptions, soon we will be able to use text files.

We have other tools that will convert TextGrid and Transcriber files and will integrate this in the near future. Please let us know about your own file formats so we can include them in future versions!

- c) Filenames of the transcription must match the audio filename.

We are working on different ways to deal with this but for now, these are best done manually.

Transcriptions don't need to be word level. Annotations at an utterance/phrase level are fine.

Clean your transcriptions by looking through them and checking the following:

- Standardise variation in spelling
- Replace non-lexical number forms, shorthand forms and abbreviations with full lexical forms. For example, replace '9' with 'nine'.
- For more cleaning tips, see the [Data preparation](#) wiki page.

You can also add text files that contain words in the language, that don't have matching audio. These will be used to improve the system's language model.





## PREPARING FILES

Speech recognition systems train on pre-transcribed speech data, building a statistical model of speech, which can then be applied to untranscribed speech. It is important to recognise that the type of speech that the system trains on will determine the type of speech that the trained model can be used on. For example if you train a system with speech of a person counting numbers, that model will be great for automatically transcribing more speech of that person counting, but wouldn't be practical for transcribing a different person telling stories.

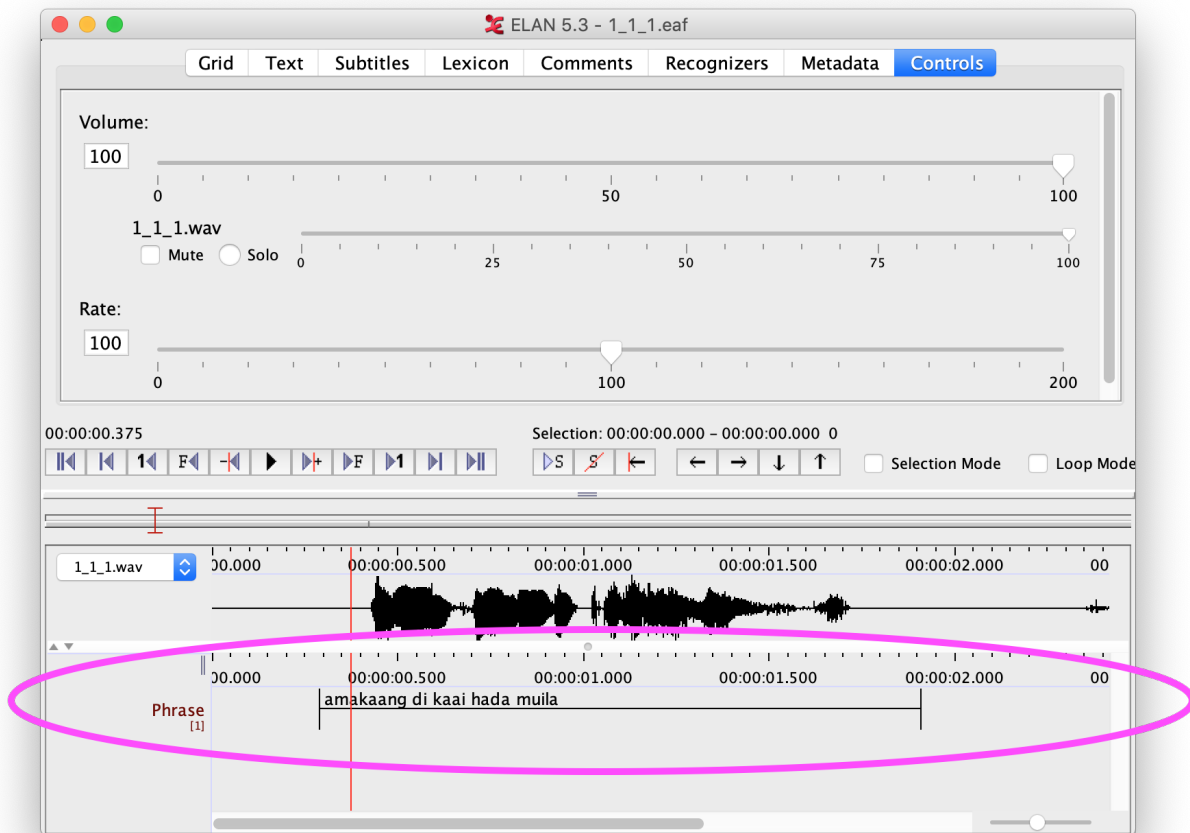
Select a collection of data for which you have the most orthographically-transcribed, high-quality speech recordings. The system will learn from these to build a model for the language.

### 14.1 Preparing your existing transcriptions

Choose a set of data from your corpus. For maximum success in the workshop, use orthographically-transcribed content from a single speaker. Select data from a common recording activity, e.g. short sentences, or stories. The current version of Elpis uses Elan files. Let us know if you have other transcription file formats, as we are currently working on adding other transcription formats.

Duplicate your data set so that you don't affect your original data by preparing for this workshop, as some of the workshop steps are destructive.

Identify which Elan tier the transcriptions are on that you want the system to learn, and ensure that this "target" tier is named consistently across all the files in this corpus.



## 14.2 Clean your transcriptions by looking through them and checking the following:

- Reduce inconsistencies or typos in transcriptions.
- Standardise variation in spelling.
- Replace non-lexical number forms, shorthand forms and abbreviations with full lexical forms. For example, replace '9' with 'nine'.
- Code-switching in a single tier will confuse the system. Although it is possible to train a multi-lingual system, in this workshop we will focus on one language. Separate multiple languages by creating one tier for the language you want to train.
- Out-of-vocabulary words (words that are in the corpus but not in the lexicon) will reduce the accuracy. Ensure that everything in the speech signal is transcribed.
- Remove inline conventions such as speaker or language codes.
- Remove punctuation that is not lexically significant

## 14.3 Audio

Elpis trains using 16 bit, 16kHz, mono WAV format audio. It will convert WAV files to the required specification, however converting your audio to these specifications beforehand will reduce the training time.

- 16 bit is the bit-depth, the number of values in each audio sample.
- 16kHz is the sample rate, also known as the sample frequency.
- Mono refers to the audio having only a single channel, rather than stereo being two channels.

Ensure the audio is in WAV format. MP3 is not suitable because the MP3 compression removes much of the information in the audio signal. Converting from MP3 to WAV doesn't work either, as the information lost in the compression is not recovered in conversion.

Audio filenames should match the transcript filenames, bearing in mind that any commas, spaces, pipes, etc. in the filenames will cause problems. Rename your filenames to only use alphanumeric with underscores or dashes.

Use noise reduction techniques to clean the audio signal if needed.

## 14.4 Examples

For examples, refer to the [Abui toy corpus](#).



## VIEWING ELPIS TRAINING LOG FILE

During the training stage a log file is written to `/state/of_origin/models/XXXX/train.log`. Note that this file isn't created as soon as the **Start Training** button is clicked, there is a slight delay while some data prep is done. To view this file:

1. Connect to the VM if Elpis is running on a cloud machine. [Install gcloud](#) if you don't have it already.

```
gcloud compute instances list
gcloud compute ssh instance-1
```

1. Run this command to get into the Elpis container.

```
docker exec -it elpis bash
```

1. Look in the `/state/of_origin/models` directory. The hashes are directories of the models that have been made. Change into the current model dir.

```
cd /state/of_origin/models
ls
cd <some_model_hash>
```

1. If `train.log` file is there, you can look at it for some insight into what Kaldi or HFT are doing. If Elpis is currently training, use `tail` to stream the log as it updates.

```
tail -f train.log
```



## HOW TO SETUP REPOSITORIES TO DEVELOP ELPIS

This guide can assist in setting up directory structures to load repositories into a Docker container, enabling you to develop code and interact with the changes. This guide doesn't cover testing.

The recommended folder structure is to have a `~/sandbox` folder inside your user directory. This can contain the `elpis` Git repository and a `state` folder to view the dataset, model and transcription sessions that the program generates.

This guide assumes the use of `zsh` rather than `bash`.

### 16.1 Prepare your local dirs

Set up a `sandbox` folder in your home directory. Create a `state` folder in there. This will be shared into the Docker container when we run it soon.

```
mkdir ~/sandbox  
cd ~/sandbox  
mkdir state
```

Clone the Elpis repo into the sandbox

```
git clone https://github.com/CoEDL/elpis.git
```

### 16.2 Build the GUI

The Docker container has a build of the React app GUI in it. If you are cloning the `elpis` repository and working on the GUI, run these commands to enable changes to the GUI code to be reloaded in the browser.

```
cd ~/sandbox/elpis/elpis/gui  
yarn install && yarn watch
```

## 16.3 Mount local dirs into existing image.

Run the Elpis Docker image. Mount your local repositories into the container. Leave out the mounts you aren't actively developing. Thus you get to use the venv in the Docker container, don't need to set up your own, avoiding version issues.

```
docker run --rm -it \  
  --name elpis \  
  -p 5001:5001/tcp \  
  -p 6006:6006/tcp \  
  -v ~/sandbox/state:/state \  
  -v ~/sandbox/elpis:/elpis \  
  --entrypoint zsh coedl/elpis:latest
```

Run this command to start the Elpis interface.

```
export FLASK_APP=elpis && flask run --host=0.0.0.0 --port=5001
```

You can also simply use the alias command inside the container.

```
run
```

## 16.4 Command-line window

elpis uses poetry for dependency management and packaging. Starting up the virtual environment might be useful if you want to develop in an IDE or text editor with autocompletion & other fancy stuff, or if you would like to run tests.

```
cd ~/sandbox/elpis  
poetry shell  
poetry update
```

## 16.5 Monitor the app/code

Open a new Terminal and get another window into the running Elpis container using this (this works on Mac, untested on PC)

```
docker exec -it $(docker ps -q) zsh
```



## BUILDING THE ELPIS DOCKER IMAGE

To build an image locally, use this command.

```
docker build --platform linux/amd64 --tag IMAGE_NAME .
```

For example,

```
docker build --platform linux/amd64 --tag coedl/elpis:latest --tag coedl/elpis:0.96.8 .
```

After building, push to the hub. Login first

```
docker login
docker push coedl/elpis:latest
docker push coedl/elpis:0.96.8
```

Or push all tags.

```
docker image push --all-tags coedl/elpis
```

### 17.1 Build issues

Docker build issues may be due to Docker storage being full. This may be indicated by error messages such as:

```
E: The repository 'http://archive.ubuntu.com/ubuntu focal-updates InRelease' is not
↳ signed.
W: GPG error: http://archive.ubuntu.com/ubuntu focal-backports InRelease: At least one
↳ invalid signature was encountered.
```

Try cleaning space with these commands, and then rebuild.

```
docker image prune
docker container prune
docker builder prune
```



## MAKING A RELEASE

Follow these steps to make a release and new Docker image for Elpis.

### 18.1 Update docs

Update docs if required with any description of changed functionality. Pushing to master will rebuild the readthedocs repo.

### 18.2 Try the code

Test GUI build as a sanity check that the app builds and eslint is happy. You may need to set your local Node to the version required for the GUI. These commands use *asdf* to manage Node version. Information about installing and using *asdf* are [here](#).

```
cd ~/sandbox/elpis/elpis/gui
asdf local nodejs 15.14.0
yarn install && yarn build
```

Build a new Docker image.

```
cd ~/sandbox/elpis
docker build --tag elpis-latest-test .
```

Check that Elpis runs with the new image. The regular `docker run ... command`, used when developing, mounts a local state directory and a local copy of the Elpis repository for developer convenience. The following command doesn't mount a state directory or local copy of the Elpis repository. This ensures the Docker container doesn't unintentionally include other libraries which may have been installed locally during development.

```
docker run --rm --name elpis -p 5001:5001/tcp -p 6006:6006/tcp elpis-latest-test
```

Open <http://0.0.0.0:5001> in a browser (or, try <http://localhost:5001> if that doesn't work), and train and test with at least a [toy corpus](#). For Kaldi, use the Abui toy corpus. The Na toy corpus may be more suitable for checking the HFT engine.

## 18.3 Update the version number in the code

Update the changelog and version details in the Elpis code, and push a commit for the version bump to master (or make a PR).

```
~/sandbox/elpis/CHANGELOG.md
~/sandbox/elpis/pyproject.toml
~/sandbox/elpis/docs/conf.py
~/sandbox/elpis/elpis/gui/package.json
```

## 18.4 Make a Git release

- Draft a [new release](#)
- Click Choose a tag and type the next version num including a leading v. E.g. `v0.96.10`
- Select + Create new tag: xxx on publish to save the tag when the release is published.
- Leave release title empty to use the tag as the title.
- Write a description of the release (should be the same as the changelog info).
- Click Publish release. This will bundle the code as `.zip` and `.tar.gz` assets with the release.

## 18.5 Docker image

Docker image builds are automatically triggered by any commit to master, which will build a Docker image tagged “latest”. Version releases will build images tagged with the version number.

The following commands can be used to build images with custom tags.

```
docker login
docker tag elpis-latest-test coedl/elpis:custom-tag
docker push coedl/elpis:custom-tag
```

Clean up Docker

```
docker image rm elpis-latest-test
docker image prune -a
```

## USING THE CLI ELPIS PYTHON API

Requires Docker.

### 19.1 Prepare your data

Make a local directory with your data, including your training data, a letter to sound file if using Kaldi, and an untranscribed audio file if you are also transcribing.

See example at <https://github.com/CoEDL/dev-corpora>

```
~/Desktop
├── datasets
│   └── abui
│       ├── letter_to_sound.txt
│       ├── transcribed
│       │   ├── 1.eaf
│       │   ├── 1.wav
│       │   ├── 2.eaf
│       │   └── 2.wav
│       └── untranscribed
│           └── audio.wav
```

### 19.2 Run Elpis, train and transcribe

Start Docker.

Run an Elpis docker container, sharing your local datasets directory with the container.

```
docker run --rm -it -p 5001:5001/tcp -v ~/Desktop/datasets:/datasets --entrypoint /bin/
↪zsh coedl/elpis:latest
```

In the container, set up a virtual environment and install dependencies. Then run the sample Python training script. It will look in the folder you gave, train a system based on the audio and text files it finds, and transcribe the untranscribed audio.

Base your own training script on the example script.

```
cd /elpis
poetry shell
python elpis/examples/cli/kaldi/train.py
```

Or, use a trained model to transcribe some audio.

```
python elpis/examples/cli/kaldi/transcribe.py
```

## USING THE CLI FOR ELPIS GPU

**This doc refers to WIP on hft branch.**

See the example script in `elpis/examples/cli/hft/train.py` for a simple demo of how to do data prep and training.

The example script will use the bundled TIMIT data that is included in the `datasets` dir in the docker container root.

The `hft` image comes with `abui`, `na` and `timit` data provided. See below for layout. To use your own data, add it as a subdir of `/datasets` and change the script `DATASET_DIR` value to suit. (To get your data into the container, use `curl/wget/git clone`.)

Running the sample script will prepare/normalise the data and do training. Subsequent runs of the script will reuse the prepared dataset, saving having to prepare the data repeatedly, and make a new training session.

If you prefer to redo data preparation, edit the Python script and change the `DATASET_NAME` value, while retaining the `DATASET_DIR` value.

Once training is complete, the model files will be in a `/state/models/XXXXX` dir.

### 20.1 Commands

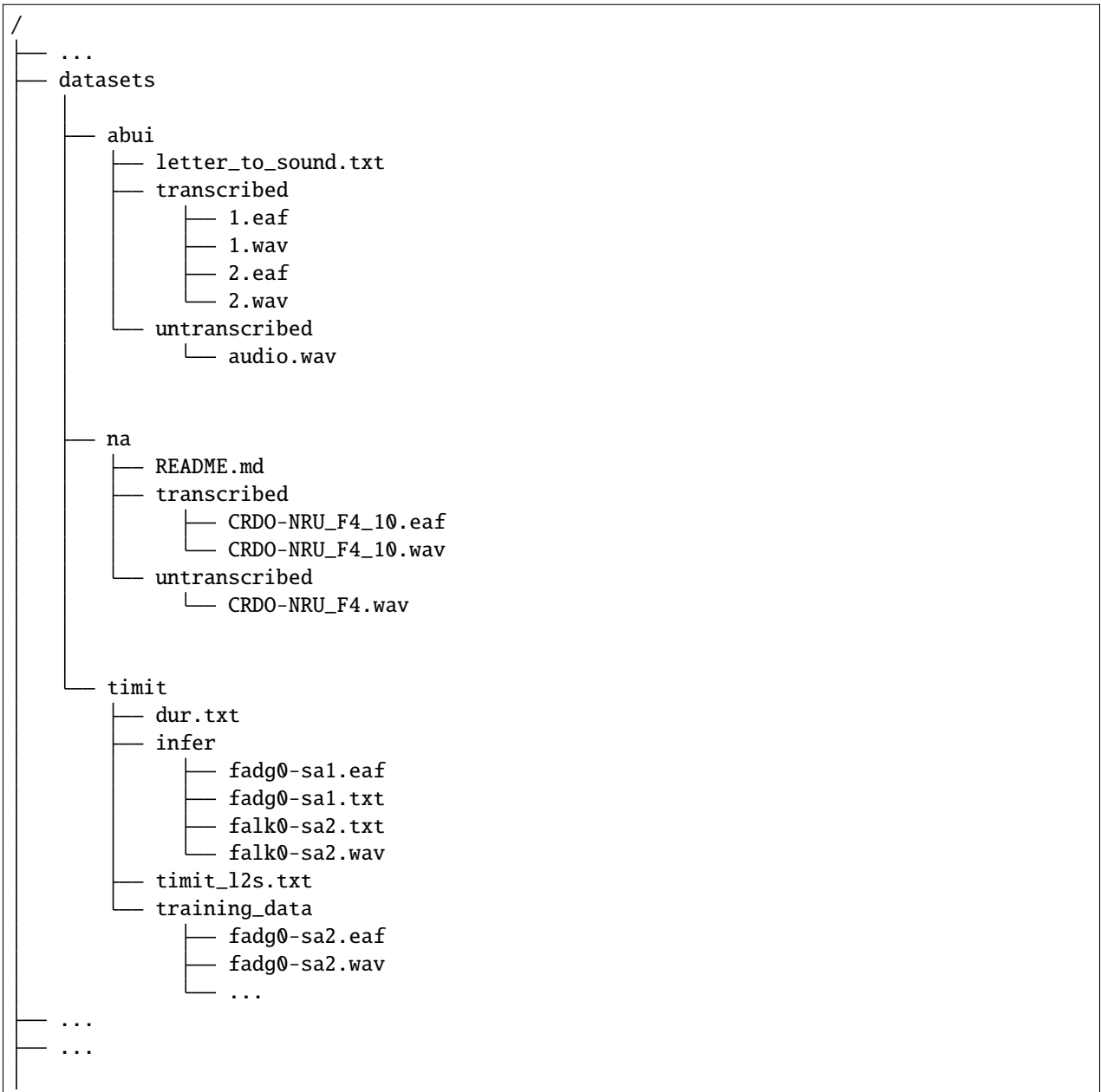
Connect to a GCP instance and run Elpis docker.

```
gcloud compute ssh instance-3
screen
docker run --gpus all --rm -it -p 80:5001/tcp --entrypoint /bin/zsh hft
git pull
python elpis/examples/cli/hft/train.py
```

To manipulate training params, open another terminal and modify the `model.py` file.

```
gcloud compute ssh instance-3
docker exec -it $(docker ps -q) zsh
vim /elpis/elpis/engines/hft/objects/model.py
```

## 20.2 Data layout





## HANDY GCP COMMANDS

Follow [these instructions](#) to install the `gcloud` tool.

### 21.1 Connect to a Virtual Machine

Use `gcloud` to connect from a local terminal to a Google Cloud Platform Virtual Machine. `gcloud init` will authorise `gcloud` to use your credentials to access your account. Then we will list the available machines, and make an SSH connection to one. Change `instance-1` in the code below to match the name of the machine you want to connect to.

```
gcloud init
gcloud compute instances list
gcloud compute ssh instance-1
```

### 21.2 Using `screen`

Using `screen` will avoid long-running training processes from terminating due to network connection failures between your machine and the VM.

Start `screen`.

```
screen
```

Do things, e.g. run a Docker container...

Then detach or reattach after a network failure.

- `Ctrl-a + Ctrl-d` to detach from the screen
- `screen -ls` to list screens
- `screen -r` to reattach

## 21.3 Copy model files from GCP

SSH to GCP instance.

Get into the docker container.

```
docker exec -it elpis zsh
cd /state/of_origin/models
ls
tar -cvf model.tar HASH_DIR_NAME
```

Keep that Docker container running, and in another SSH terminal, copy from the container to the host.

```
docker cp elpis:/state/of_origin/models/model.tar .
```

Copy from the host to the local machine (do this in a local terminal window).

```
gcloud compute scp instance-1:~/model.tar ~/Downloads/model.tar
```

Otherwise, could share state dir from host into docker and save a few steps...

## 21.4 Fixing the SSH Key

When making an SSH connection using gcloud, you may receive a `Remote Host changed` error. This can be fixed by regenerating some files on your computer.

Use the Google Cloud Console in your browser to check that the VM is running.

Delete these files from your computer.

```
~/.ssh/google_compute_engine
~/.ssh/google_compute_engine.pub
~/.ssh/google_compute_known_hosts
```

Run these commands on your computer to authorise you and generate new SSH keys. Replace the zone, instance and project names to suit your situation.

```
gcloud auth login
gcloud compute ssh --zone "us-central1-c" "instance-1" --tunnel-through-iap --project
↪ "elpis-workshop"
```