The Open Handbook of Linguistic Data Management

Edited By: Andrea L. Berez-Kroeker, Bradley McDonnell, Eve Koller, Lauren B. Collister

Citation:

The Open Handbook of Linguistic Data Management
Edited By: Andrea L. Berez-Kroeker, Bradley McDonnell, Eve Koller, Lauren B. Collister
DOI: 10.7551/mitpress/12200.001.0001
ISBN (electronic): 9780262366076
Publisher: The MIT Press
Published: 2022

The MIT Press

36 Managing Transcription Data for Automatic Speech Recognition with Elpis

Ben Foley, Daan van Esch, and Nay San

1 Introduction

This chapter provides a mid-level introduction to speech recognition technologies, with particular reference to Elpis (Foley et al. 2018), a tool designed for people with minimal computational experience to accelerate their language documentation transcription workflows by taking advantage of modern speech recognition technologies (e.g., Kaldi [Povey et al. 2011] and ESPnet [Watanabe et al. 2018]). By a *mid-level introduction*, we mean that the chapter focuses on the *whats* and *whys*, rather than low-level *hows* (e.g., click button X, then . . .); such how-to descriptions are provided via the latest stable version of the Elpis project's documentation (Foley, Lambourne, & San 2020). On the other hand, for a more comprehensive, high-level introduction to speech recognition models, we recommend Jurafsky and Martin (forthcoming).

Elpis is intended to be used in situations where there might not be the large quantities of already transcribed recordings typically required for training commercially viable speech recognition systems (which are usually trained on hundreds to thousands of hours of transcribed recordings). Even in language documentation contexts where people may only have one or two hours of transcribed recordings, using speech recognition can be beneficial to the process of transcription. A speech recognition system can be built using small quantities of recordings and used to generate a rough "best guess" for untranscribed audio. This new transcription can be corrected and used to retrain the system, improving the quality of the rough transcription with each iteration.

Moreover, in becoming more familiar with the types of data and metadata necessary to train a speech recognition system (as well as best practices for their organization), language documentation teams may develop a better-informed data management plan that facilitates the adoption of semi-automated workflows.

In the first section, we describe the motivations for this work, provide an overview of various components of a speech recognition system, and describe their roles within the overall system. Subsequently, we discuss how to plan and prepare data for use with Elpis. The technologies discussed here are designed to learn from acoustic features of spoken language and are not effective for signed languages.

1.1 Motivations

The motivations for developing Elpis are two-fold. The first is to further amplify the many ongoing efforts of people transcribing recordings of spoken language, particularly endangered and under-resourced languages. The second is to bring the possibility of using speech recognition technologies, which are currently restricted to very few languages and inaccessible for people without specialist training, to many more of the world's languages.

A transcription is the textual representation of language, typically made by writing while listening to or watching a recording. Transcriptions can be created by typing into a text file, by using software such as ELAN (ELAN 2019), or even by writing by hand on paper. When done from scratch without assistance from automatic speech recognition software, transcription tends to be slow—a survey of linguists in 2017 reported an average of forty minutes taken to transcribe one minute of speech (Foley et al. 2019). While digital technologies make it easy to record large quantities and varieties of knowledge, this bottleneck of transcription limits the amount that may be turned into written text.

Indeed, this transcription bottleneck has long been recognized (Himmelmann 2018). With recent advances in the field of automatic speech recognition, there has been an increasing amount of interest in employing speech recognition tools in language documentation contexts, and various interdisciplinary collaborations are reporting promising results. For example, Michaud et al. (2018) describe the positive change in workflow experienced by using Persephone for phonemic transcription of Na language. Seneca transcribers report approximately 40% reductions in both the time taken and transcription word error rate when using the Kaldi speech recognition tool kit (Jimerson et al. 2019).

However, a widespread uptake of state-of-the-art tool kits such as Kaldi and ESPnet for use on many of the world's languages is hindered by two primary barriers. The first has traditionally been the general availability of adequate quantities of recordings and transcriptions to train speech recognition systems. With many language communities now proactively making their own language recordings and crowdsourcing the transcription work, this first barrier has been substantially lowered for many languages.

Still, the specialist expertise required to design, build, or even run an existing speech recognition system are beyond the reach of many. Elpis aims to lower the barrier of entry to help people work toward implementing speech recognition systems for their own language by providing an easy-to-use interface. In particular, Elpis was designed with language workers and linguists to provide a graphical user interface to assist the preparation of files for training and recognition using Kaldi for orthographic transcription. Elpis has since been extended to provide an accessible interface for phonemic transcription using ESPnet.

1.2 Overview

Automatic speech recognition (ASR) is a technology used to generate a written transcription of an audio signal. In general, the term is used to refer to a process that involves stages of preparation of existing recordings and transcriptions, using this existing data to train the system, and then using the trained system to infer an automatic transcription for untranscribed audio (this automatic transcription is called the *machine hypothesis*). ASR is used in applications such as speech interfaces on devices such as Google Assistant, Amazon Alexa, or Apple Siri, but beyond such consumer use cases, it can also be beneficial for assisting in workflows of transcribing speech recordings (Michaud et al. 2018).

For orthographic transcription, Elpis uses a statistical ASR system to determine the likelihoods that acoustic units occur in particular sequences, based on the acoustic and textual information provided in the training data. Currently, the default method used is based on a combination of hidden Markov models (HMMs) and Gaussian mixture models (GMMs). The decision to use the HMM+GMM–based technique is based on the attributes of the recordings that the users involved in the first stage of design of Elpis had access to: collections of recordings less than tens of hours in total duration. In general, for languages with large quantities of training material, neural network techniques perform better. However, for small quantities of recordings, HMM+GMM systems are computationally efficient and have comparable performance to neural networks. In future, neural networks for orthographic transcription will be added to Elpis to benefit users with larger quantities of training data.

For phonemic transcription, Elpis uses ESPnet (Watanabe et al. 2017), another speech processing tool kit that is compatible with Kaldi-style input data (and hence can be used with the data preparation procedures Elpis facilitates).

ESPnet provides an "end-to-end" approach, which generally requires more data though reduces the need to prepare "hand-crafted" intermediary data, such as a pronunciation lexicon (introduced below).

Specifically, ESPnet in Elpis uses a hybrid CTC-attention model (Watanabe et al. 2017) with a three-layer BiLSTM encoder and a single layer decoder.

Elpis can be installed on local computers or networked servers, suiting the needs of different users. By running entirely on local hardware, such as a laptop or a desktop, recordings do not have to be uploaded to the cloud, which would pose a privacy and data sovereignty concern for many language communities (Holton, Leonard, & Pulsifer, chapter 4, this volume).

Each ASR technology has data preparation needs that differ according to the requirements of the system. In the following sections, we describe how to prepare language recordings and transcriptions to use with Elpis.

2 Components: Acoustic model, language model, pronunciation lexicon

In this section, we describe the components of the Kaldi orthographic transcription system, which uses a statistical method. For details of the phonemic transcription system used in Elpis, see Adams et al. (2020).

Statistical ASR systems are typically composed of three main parts: an acoustic model, a pronunciation model, and a language model. These models are trained or built separately using some previously transcribed audio and text. The models are then combined and used to provide a hypothesis about the text representation of untranscribed audio.

The acoustic model aims to produce a phonemic transcription for a given input audio recording. It is trained by providing speech recordings along with an orthographic transcript of each utterance. This orthographic transcript is turned into phonemes at training time, and the model is trained by feeding in segments of the waveform along with the appropriate phoneme labels to output.

The pronunciation model prepared by Elpis is simply a pronunciation dictionary, a list of all the words in the lexicon and a representation of how each word is pronounced. These representations are typically phonemic transcriptions. Depending on the degree of orthographic transparency of the target language, these phonemic transcriptions can be straightforwardly derived from the orthography through letter-to-sound rules, or they may need to be curated manually on a word-by-word basis.

The language model is a statistical representation of the occurrence of word sequences in the language. It is trained on the transcription text but may also additionally be trained on any other text-only material that may be available. When combined at runtime, the acoustic model first provides a set of best guesses as to the series of phonemes contained within a previously unseen snippet of audio. The pronunciation model then turns these phonemic transcription candidates into a list of word candidates. The language model then helps further disambiguate these candidates according to what it knows are the most likely series of words that tend to follow each other (learned from the training data).

When a system is trained on small quantities of recordings, typically with few speakers, it will not generalize well to new speakers. In the language documentation context for which Elpis was designed, it is rare to have access to large quantities of recordings to build general conversation systems that can scale for many speakers. However, in the context of language documentation with few speakers, scaling can be less of an issue if most of the speakers are previously observed. It is also important to acknowledge that the resulting system is somewhat limited in its ability to transcribe audio with very different characteristics: for example, if there is little background noise in any of the original audio recordings, the system will struggle to transcribe new recordings with significant amounts of background noise. Another limitation is that the trained system can generally only emit words that have been observed in the original training data, either in the text-and-audio pairs, or in the text-only supplemental materials.

3 Using Elpis

The process of using Elpis for either orthographic or phonemic transcription begins with planning and organizing your recordings and transcriptions. Cleaning, normalizing, and standardizing the transcription text is critical, and effort spent here will make a positive impact on the performance of the system. For detailed steps, refer to tutorial information provided in Foley, Lambourne, and San (2020).

Elpis can be used either locally on high-end laptops or desktop computers or as software on a cloud server. When using Elpis locally, no network connection is needed, making it ideal for fieldwork use in areas with limited connectivity. Local use can also be helpful in enabling the use of automatic speech transcription in situations where privacy/security requirements or data ownership concerns make uploading recordings to cloud services for transcription infeasible. Using a cloud version of Elpis can provide access to large amounts of computing power, reducing the time it can take to train the system. It may also be possible to use local university-run computer clusters to run Elpis. Choosing which method of running Elpis will depend on the particular needs of your language.

3.1 Workflow

A typical workflow begins with making recordings and manually transcribing some of the recordings. For orthographic transcription, a file that maps letters in the orthography to symbols representing how they are pronounced is also prepared. This file is not required for phonemic transcription. The recordings, transcriptions, and letter-to-sound file would then be added to Elpis. For orthographic systems, the pronunciation dictionary is generated by Elpis, and the results are checked and edited. Then the models are trained. After training, untranscribed audio can be passed through the trained model to obtain a hypothesis transcription. The transcription can be viewed and downloaded as a text or ELAN file.

Currently, using Elpis is a linear workflow, requiring retraining when new training data is added to the corpus. For example, if we train a system with three hours of data, we could obtain a hypothesis for fifteen minutes of untranscribed audio and edit the hypothesis. The new transcription can then be added to the corpus. We would then create a new collection of files in Elpis. The pronunciation dictionary may need to be regenerated if there are new words in the new data, and the training would be rerun. This would be repeated for each new, edited hypothesis we wanted to include in the training corpus. Adding data into the training corpus after editing is not mandatory, but it allows the system to learn and potentially correct similar mistakes when transcribing more audio.

3.2 Planning and preparing data

The quality of results that we can obtain from an ASR system depends on the quality and quantity of the recordings and text used to train the system. To build an ASR system's acoustic model, audio is analyzed to learn various examples of each phoneme, or unit of speech. The clearer the recordings that are provided, the better clarity the system will have in "hearing" the difference between phonemes. High-quality audio is said to have a high signal-to-noise ratio, meaning that the voice signal is clear and there is little interfering background noise. Other significant factors include managing multiple people speaking during the activity and repetition of utterances.

Before adding the transcription files to Elpis, the transcriptions must be cleaned to correct typographic errors, normalize orthography, and remove non-lexical text and codes from the annotations. Cleaning transcriptions can take time, but the effort put into cleaning will improve the results of the system. When starting a new language documentation project using an ASR system, make sure to pay careful attention to the consistency of the transcriptions.

For an example of the data used to train Elpis refer to the Abui toy corpus,[1] a tiny selection of transcriptions from a larger corpus of a Papuan language of Alor Island, Indonesia (Kratochvíl 2007), which has been cleaned and prepared according to the steps outlined herein.[2]

3.2.1 Planning for clean recording Audio quality can be affected by many factors ranging from the location in which recording happens, the social activity surrounding the recording event, the types of equipment used to make the recording, and the formats used in storing the recordings. Thus, when planning a recording session, try to use a location in which ambient noise can be controlled, such as a room in which air-conditioners can be turned off, or locations that minimize surrounding animal or traffic noises. When recording outside, the impact of wind on a recording can be minimized by using wind shields on microphones, putting up barriers to block wind, or recording in portable structures such as Green's signadome (Green, Woods, & Foley 2011); structures such as the signadome also provide shade for participants, keep out wayward dogs and children, and focus participants' attention on the recording activity at hand. For a detailed guide to planning and recording a language documentation activity in a fieldwork context, see Meakins, Green, and Turpin (2018).

3.2.2 Managing multiple speakers When planning a recording activity for ASR with multiple speakers, two key factors will affect the ability to use the recordings: separating speakers as audio sources, and overlapping speech.

Directional head-mount microphones with multitrack recording devices are ideal for isolating speakers to individual tracks. Where possible, provide each speaker with a microphone and record to separate tracks on the recording device. This will enable individual speakers to be more easily used in training a multiple-speaker system, by keeping each speaker's voice unmixed. Elpis currently cannot separate the acoustic signal of speakers when they talk over each other. Prior to recording, encourage speakers to be conscious of talking over the top of others and making back-channel interjections. Minimizing the occurrence of overlapping speech during the recording will lead to a cleaner acoustic signal for training.

3.2.3 Repetition The more instances of each phoneme that can be provided, the more opportunities the ASR system will have to learn, and consequently the ASR system will be more robust. When recording an elicitation activity, try to have a few utterances that are repeated a few times, and aim for some variation in what is recorded, for example, slightly faster and slightly slower speech, some expressive prosody, and some flatter delivery. For multiple speaker systems, try to record repetitions by each speaker. Supply all of these recordings (along with their matching transcriptions) as training data to expose the system to variations in the sounds.

3.2.4 Audio format When recording, use a lossless format such as WAV rather than a format such as MP3 that discards and compresses audio information (Mattern, chapter 5, this volume). Set the device to record at the highest setting it is capable of—contemporary recording equipment will typically record at 48 kHz or higher. Elpis expects mono (single-channel), 16-bit sample depth, 44.1 kHz sample rate, WAV format audio. If the supplied audio is different to these specifications, Elpis will convert the source to match. Note that it is good to record at higher sample rates and allow Elpis to downsample, as this will provide more flexibility for future reuse of the recordings.

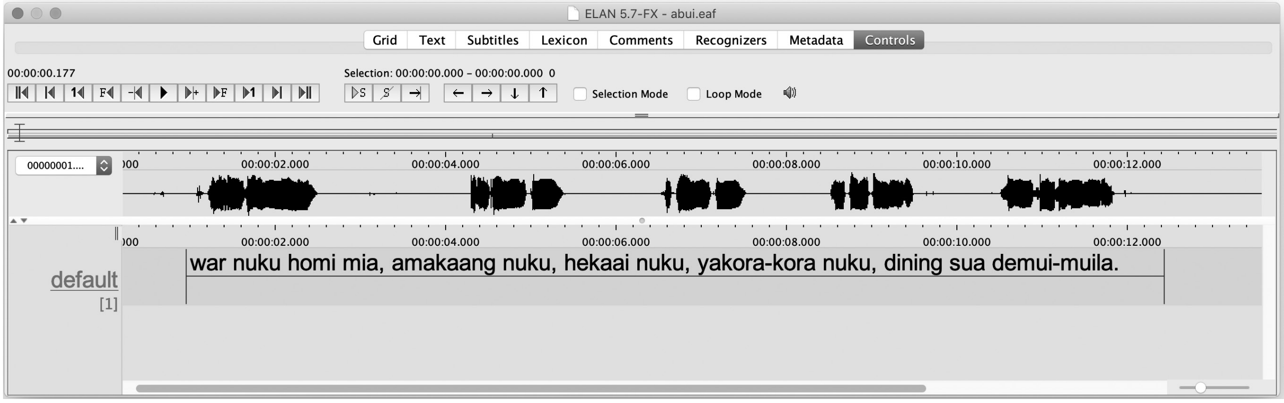
3.2.5 File handling It is good practice for language recordings to have unique file names, which will ensure that recordings can be uniquely identified. It is particularly important for Elpis that file names are unique across the corpus, so as to prevent erroneous associations of transcriptions to audio when the audio and transcription files are uploaded and added to the collection of training files.

Elpis requires that the base names (the file name part before the period and extension) of each file in a pair of audio and transcription match, with a differing extension, for example: audio1.wav, audio1.eaf, audio2.wav, audio2.eaf. Prior to your recording activity, set your recording device to use a file-naming template, following good data management planning practices such as those mentioned in Kung (chapter 8, this volume). If the recording device doesn't have a naming template, rename the files after getting them off the device. Rather than doing this manually, use batch-renaming software to save time.

3.2.6 Annotation segmentation Elpis requires the existing audio and transcriptions to be aligned at an utterance level, meaning that it is sufficient to transcribe in chunks or segments that are approximately ten seconds in duration or less (see figure 36.1). Word- or phoneme-level segmentation of the training data is not necessary.

3.2.7 Typographic errors Check the transcriptions for typographic errors. For now, this is a manual process that is best done using the software that the transcriptions are written with; however, planned Elpis work includes the ability to edit the transcripts directly in the interface. Pay attention to spelling errors, and for consistency in spelling across the corpus, especially for transcriptions that may have developed over many years as the transcribers' knowledge of the language evolved.

3.2.8 Orthographic consistency To maximize the frequency of occurrence of words, consistency in orthography within the training data is important. Spelling the same word in multiple ways will confuse the language model and dilute the frequency count for that word among the number of spellings used. The challenge for many languages is that there may not be a standard orthography, or the orthography may change over time



war nuku homi mia, amakaang nuku, hekaai nuku, yakora-kora nuku, dining sua demui-muila.

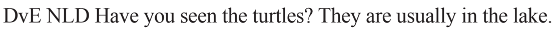
Figure 36.1
Annotation segment in ELAN approximately ten seconds in duration.

in response to changes in social-political and/or linguistic influences. Even in these contexts, developing a *working orthography* just used for training Elpis, and conforming your training transcriptions to this orthography will benefit the system (even if some speakers and/or language experts do not agree on adopting this orthography as a broader standard).

3.2.9 Faithful transcriptions Ensure that everything in the speech signal is transcribed faithfully and that only human speech is transcribed. For example, if the speaker says 'hello hello', then both *hellos* should be transcribed. Likewise, transcriptions should not include references to non-speech sounds. If dogs are barking, or someone is laughing in the audio, do not transcribe as "dogs barking" or "people laughing"; just leave the descriptions out or make an annotation describing the audio on a different tier if using ELAN.

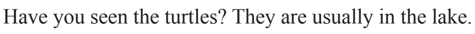
3.2.10 Non-lexical forms Non-lexical numbers and shorthand forms of words should be replaced in the transcriptions with full lexical forms. For example, replace "9" with "nine"; if someone says 'for example', check that the transcript reflects what is said, and is not written as "e.g." Likewise, for abbreviations, check that the transcription reflects what was spoken. This cleaning stage is an important step in ensuring that what is written is as close as possible to the acoustic signal.

3.2.11 Speaker and language codes Other common transcription artifacts that are among the transcription text, such as speaker and language codes (see figure 36.2), should be removed (see figure 36.3) and kept as tier attributes, in separate transcription tiers (if using ELAN), or in metadata files.



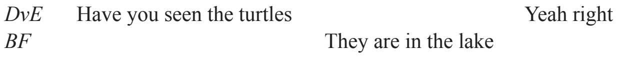
DvE NLD Have you seen the turtles? They are usually in the lake.

Figure 36.2
Original transcription has speaker (DvE) and language codes (NLD).



Have you seen the turtles? They are usually in the lake.

Figure 36.3
Speaker and language codes removed from the transcription.



DvE Have you seen the turtles Yeah right
BF They are in the lake

Figure 36.5
Tiers for each speaker.

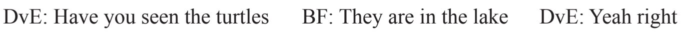
Although it is possible to train multilingual ASR systems, Elpis is currently configured to train single-language systems. If the recordings contain multiple languages, separate the languages to different tiers (if using ELAN) or delete the second-language transcriptions from the data. When removing languages from the transcriptions, work with a copy of the data, as this is a destructive process.[3]

Similarly, for multiple speakers, rather than including speaker codes in a single tier (see figure 36.4), annotate each speaker on a separate tier, labeled with the speaker's name, or add the speaker's name as a tier attribute (see figure 36.5) (see sentence figure 36.4).

3.2.12 Automatic cleaning After thoroughly cleaning and normalizing the audio and transcription files, they can be added to Elpis. When they are added, some further processing stages occur, including automatic removal of English words and other text that you can specify, stripping punctuation, and changing the corpus to lowercase. Elpis uses the Natural Language Toolkit (NLTK) English corpus to identify words in the transcriptions that match and removes them. For utterances that are greater than 10% English in total, the whole utterance is excluded from the data. Elpis will remove certain punctuation marks, including periods, commas, single and double quotation marks, and so forth. The set of punctuation marks to be removed can be customized by the user.

3.3 Pronunciation lexicon

For orthographic transcription, a plain text file that lists the characters used in the transcription, along with symbols that represent the pronunciation, is used to create a pronunciation lexicon. The text file is also referred to as grapheme-to-phoneme (G2P), letter-to-phoneme, or letter-to-sound mapping. The letter-to-sound mapping in Elpis is crude, intended to be a rough draft that can be improved iteratively.



DvE: Have you seen the turtles BF: They are in the lake DvE: Yeah right

Figure 36.4
Original single-tier structure.

```
m m
ng ŋ
n n
r r
y j
l l
q q
v w
# vowels
uu u:
u u
ii i:
```

Figure 36.6
A section of a letter-to-sound file.

The letter-to-sound text file is prepared by listing all the characters used in the orthography in one column. In a second column, separated by a space, a symbol is placed to represent the pronunciation, such as an International Phonetic Alphabet (IPA) or X-SAMPA symbol (Elpis accepts many consistent phonemic transcription systems). Comments can be written in the file with a # starting the comment line (see figure 36.6).

After uploading the letter-to-sound file, Elpis generates a pronunciation dictionary (see figure 36.7). It does this crudely by splitting each word in the lexicon into individual characters and replacing them with the pronunciation symbol. This works better for languages where the orthography closely matches the pronunciation. Regardless, this is a stage where the results will need to be reviewed and edited.

When reviewing the results, corrections can be made directly in the interface for individual words, or system-wide changes can be made by updating the letter-to-sound.txt file. Check words that have been transcribed with consecutive matching characters. Do they represent one sound or two? If there is only one, add a line to the letter-to-sound.txt file, mapping the consecutive characters to a single symbol and rebuild the lexicon.

For example, if a word ‘singer’ is erroneously mapped to “singer s i n g ə” in the lexicon, where for Australian English it should have a single velar nasal *ŋ* instead of an alveolar nasal *n* followed by a velar stop *g*, add “ng ŋ” to letter-to-sound.txt above the entry for “n n” (see figure 36.8), upload it again, and rebuild the lexicon. The results should be collapsed lexicon entry “singer s i ŋ ə”.

```
di d I
ba b a
amakaang a m a k a: ŋ
botol b O t O l
yaari j a: r I
dining d I n I ŋ
hekaai h E k a: I
kamar k a m a r
mui m u I
dong d O ŋ
hepikaai h E p I k a: I
muila m u I l a
ayoku a j O k u
hada h a d a
mia m I a
ong O ŋ
kaai k a: I
homi h O m I
hayei h a j E I
deina d E I n a
del d E l
```

Figure 36.7
A pronunciation dictionary generated by the letter-to-sound map.

```
ng ŋ
n n
```

Figure 36.8
Specifying consecutive characters and prioritizing the order of entries.

Make sure to put these replacements earlier in the letter-to-sound file, above single characters.

When working iteratively to develop the pronunciation lexicon, incrementally name the letter-to-sound file when you make changes to it. Use the same name as the name for this step in Elpis. This way you can track which version you are using and which improvements you have made.

A dictionary can be manually created and used to train a model with other tools such as Phonetisaurus (Novak, Minematsu, & Hirose 2016), but this is not currently supported in Elpis.

3.4 Language model

The language model is a probabilistic approach to predicting the order of words occurring in the language, as observed in the training text. It is used to disambiguate the occurrence of words that have the same pronunciation, for example, identifying which is the correct option from *you're book* or *your book*. A probabilistic language model is also known as a grammar, but this is quite a different, somewhat simpler, sense of the word than a linguist's grammar.

3.4.1 N-grams For orthographic transcription, the language model is learned from sequences of consecutive words in the training files: chiefly, the audio-and-text pairs, but optionally also any supplementary text-only materials such as word lists, bible translations, dictionaries. The number of words in a sequence is referred to as an *n-gram*, with individual words being a *unigram*, word pairs are *bigrams*, three words in a row are *trigrams*, four words are *4-gram*, and so on. For example, given the sentence "The quick brown fox jumped . . . ," the language model can be trained to learn single words (unigrams) or words with their neighbors (bigrams, trigrams, and so on), as follows:

Unigrams: the, quick, brown, fox, jumped

Bigrams: the quick, quick brown, brown fox, fox jumped

Trigrams: the quick brown, quick brown fox, brown fox jumped

The n-gram value is a setting that can be chosen in the Elpis interface prior to starting the training process. The optimal n-gram value for a language will be different depending on the training transcription text used. Although larger n-gram values (larger groups of words) generate more information, in language documentation it can be difficult to record enough instances of larger structures to give statistical value to the structures, especially for languages with small quantities of training recordings.

Consider a tiny corpus with three sentences: "I like to play tennis with Daan"; "I like to play tennis with Nay"; and "I like to play football with Nay." In this tiny corpus, a 7-gram structure would have the maximum amount of information. However, we would only see a single occurrence of each instance, so a probabilistic system would not be able to choose a likelihood between the options, each being equally likely. If we used a smaller n-gram, our model would see higher occurrences of the bigrams "play tennis" and "with Nay" because there are two occurrences of those bigrams in the training text. In this way, our model would learn a higher likelihood of certain word structures, with each of those structures occurring twice compared with the lower probability of "play football" and "with Daan" only occurring once.

Unseen word sequences are determined using a technique called *back-off*, which looks for the probability of a word sequence at the previous n-gram level. For example, if we are using trigrams, and no trigram structure of a hypothesis occurs, the system can back-off and try bigram structures. If no bigrams are known, we could fall back to using unigrams. This is clearly not how the human mind works, but it is remarkably effective for machines.

3.5 Diagnostics

3.5.1 Word error rates When Elpis trains a model on a new data set, it splits off part of the data set to use as a *held-out test set*. This test set is used to evaluate the speech recognition model during and after the training process by producing an accuracy metric known as *word error rate*, or WER. This metric is computed by comparing the machine transcription of the audio in the test set to the human-provided transcription of this audio, which is assumed to be correct. Because the test set is not included in the training data, this provides a fair assessment of the level of quality that could be expected on similar entirely new, untranscribed data that is fed into Elpis for transcription. By similar, we mean data with the same characteristics, such as in terms of topic, background noise levels, and so on. In general, the closer the data characteristics are to the training data, the better the model will perform.

Lower WERs correspond to better performance: put simply, WERs reflect the number of words transcribed incorrectly within the test set, and fewer errors are better. Specifically, WER is calculated by counting three types of errors:

1. Deletions, where a word is not transcribed but it should have been. For example, 'hello world' is mistranscribed as "hello" with the word *world* missed.
2. Insertions, where a word is transcribed where no word should be transcribed. For example, 'hello world' is mistranscribed as "hello world people" with the word *people* incorrectly inserted.
3. Substitutions, where a word is transcribed incorrectly, replacing another word. For example, mistranscribing 'hello world' as "hello moon" incorrectly substitutes the word *world* with the word *moon*.

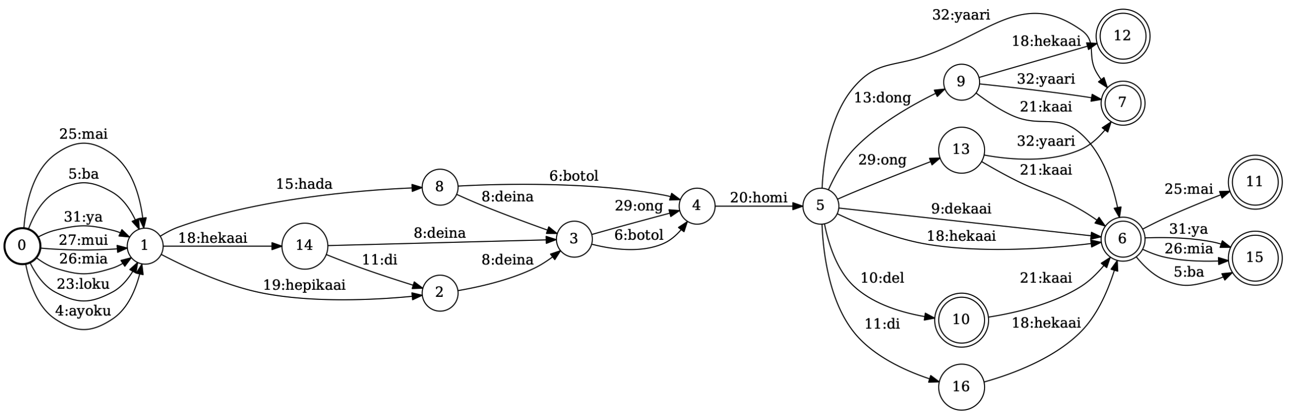


Figure 36.9
Lattice generated for an Abui utterance.

To produce the WER, the number of errors is then added up and divided by the number of words in the correct transcription. WERs will vary depending on the complexity of the language and the recordings, but in general, you should expect to see the WER drop over time as you include more training data for the models.

When performing phonemic transcription with ESPnet, Elpis reports a *phoneme error rate* (PER), an equivalent system for reporting phonemic errors rather than word errors.

3.5.2 Lattices When Elpis orthographically transcribes new audio, it runs the audio through the acoustic, pronunciation, and language models to produce a transcription. Under the hood, however, modern speech recognition systems do not generate just one transcription. Instead, they produce a *lattice* of options, which you can read as different ways the audio could be transcribed.

Reading from left to right in figure 36.9, there are multiple options for the first word, and then a number of different paths leading into different *states* of the lattice. These all represent potential transcriptions of the audio, and Elpis returns the most likely of the paths going through this lattice as the final transcription. In some cases, however, the model's second guess may actually turn out to be correct.

Up to the time of writing this chapter, the main focus of Elpis development has been on consultatively creating a user-friendly interface for data preparation for Kaldi and ESPnet. As this stage of the development stabilizes, we are experimenting with approaches to providing the diagnostic information (e.g., WER, lattices) in a user-friendly manner and in ways that can be easily integrated into typical language documentation transcription workflow patterns.

4 Conclusion

We have described the main components of statistical speech recognition systems and described what is required to prepare data to use Elpis, along with the reasons why data preparation is required. This chapter provided information in addition to what is covered in Elpis tutorials, to provide a greater contextualization of the steps involved in using Elpis.

By following the guidance given here in preparing, cleaning, and normalizing files, the value of the recordings and transcriptions will be maximized for training Elpis, other ASR systems, and other language technologies. It is often thought that only supplying greater quantities of data to a system will improve the system, but in fact, there is much to be gained (and learned) from improving the quality of existing material.

We hope that this chapter has provided an understanding of the data preparation requirements and the potential for automatic speech recognition to be used in language documentation workflows.

Notes

1. https://github.com/CoEDL/toy-corpora

2. We thank František Kratochvíl for graciously offering the Abui data for use during the first iteration of the Elpis workshop (June 28, 2017) and Elpis-related demonstrations beyond.

3. See Han (chapter 6, this volume) for considerations in planning and managing data transformation processes.

References

Adams, O., B. Galliot, G. Wisniewski, N. Lambourne, B. Foley, R. Sanders-Dwyer, J. Wiles, et al. 2020. User-friendly automatic transcription of low-resource languages: Plugging ESPnet into Elpis. *Proceedings of the 4th Workshop on the Use of Computational Methods in the Study of Endangered Languages (ComputEL)*. Vol. 1. Online.

ELAN, version 5.7 (Computer software). 2019. Nijmegen, the Netherlands: Max Planck Institute for Psycholinguistics. https://tla.mpi.nl/tools/tla-tools/elan/.

Foley, B., J. T. Arnold, R. Coto-Solano, G. Durantin, T. M. Ellison, D. van Esch, S. Heath, et al. 2018. Building speech recognition systems for language documentation: The CoEDL Endangered Language Pipeline and Inference System (Elpis). In *The 6th International Workshop on Spoken Language Technologies for Under-Resourced Languages (SLTU)*, ed. S. S. Agrawal, 200–204. https://www.isca-speech.org/archive/SLTU_2018/pdfs/Ben.pdf.

Foley, B., G. Durantin, A. Ajayan, and J. Wiles. 2019. Transcription Survey. Paper presented at the 52nd conference of the Australian Linguistic Society (ALS2019), Sydney, Australia, December 11–13.

Foley, B., N. Lambourne, and N. San. 2020. *Elpis documentation*. Zenodo. December 27. https://doi.org/10.5281/zenodo.4394816.

Green, J., G. Woods, and B. Foley. 2011. Looking at language: Appropriate design for sign language resources in remote Australian Indigenous communities. *Sustainable Data from Digital Research: Humanities Perspectives on Digital Scholarship*. http://ses.library.usyd.edu.au/handle/2123/7949.

Himmelmann, N. P. 2018. Meeting the transcription challenge. In *Reflections on Language Documentation 20 Years after Himmelmann*, ed. B. McDonnell, A. L. Berez-Kroeker, and G. Holton, 33–40. Honolulu: University of Hawaiʻi Press. https://scholarspace.manoa.hawaii.edu/handle/10125/24806.

Jimerson, R., R. Hatcher, R. Ptucha, and E. Prud'hommeaux. 2019. Speech technology for supporting community-based endangered language documentation. Poster presented at the 6th International Conference on Language Documentation and Conservation (ICLDC), Honolulu, Hawaiʻi, February 28–March 3.

Jurafsky, D., and J. H. Martin. Forthcoming. *Speech and Language Processing*, 3rd ed. https://web.stanford.edu/~jurafsky/slp3/.

Kratochvíl, F. 2007. *A Grammar of Abui*. Doctoral thesis, Leiden University. http://hdl.handle.net/1887/11998.

Meakins, F., J. Green, and M. Turpin. 2018. *Understanding Linguistic Fieldwork*. Abingdon, UK: Routledge.

Michaud, A., O. Adams, T. A. Cohn, G. Neubig, and S. Guillaume. 2018. Integrating automatic transcription into the language documentation workflow: Experiments with Na data and the Persephone toolkit. *Language Documentation and Conservation* 12:393–429. https://scholarspace.manoa.hawaii.edu/handle/10125/24793.

Novak, J. R., N. Minematsu, and K. Hirose. 2016. Phonetisaurus: Exploring grapheme-to-phoneme conversion with joint n-gram models in the WFST framework. *Natural Language Engineering* 22 (6): 907–938. https://doi.org/10.1017/S1351324915000315.

Povey, D., A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, et al. 2011. The Kaldi Speech Recognition Toolkit. Paper presented at IEEE 2011 Workshop on Automatic Speech Recognition and Understanding. Hilton Waikoloa Village, Big Island, Hawaii.

Watanabe, S., T. Hori, S. Kim, J. R. Hershey, and T. Hayashi. 2017. Hybrid CTC/attention architecture for end-to-end speech recognition. *IEEE Journal of Selected Topics in Signal Processing* 11 (8): 1240–1253.

Watanabe, S., H. Takaaki, K. Shigeki, H. Tomoki, N. Jiro, U. Yuya, N. E. Y. Soplin, et al. 2018. *Espnet: End-to-end speech processing toolkit*. arXiv preprint. arXiv:1804.00015.