Computational tools for language documentation: explorations in Automatic Speech Recognition on fieldwork data

Séminaire pratique des doctorants Llacan-Lacito

Cécile Macaire[1], Alexis Michaud[1]

cecile.macaire@live.com, alexis.michaud@cnrs.fr

July 2, 2021

[1] Langues et Civilisations à Tradition Orale (LACITO), CNRS-Sorbonne Nouvelle, France

Abstract

[French below]

The LLACAN and LACITO research centres are engaged in exploratory work that aims to tap the potential of computational methods to facilitate the documentation of endangered languages. Machine learning-based tools can effectively assist in linguistic annotation tasks, including transcription, glossing, and translation. However, Natural Language Processing tools are still little used in language documentation, mainly because the technology is still new (and evolving rapidly) and there is a lack of simple and user-friendly interfaces. Our research centres aim at co-construction of models and tools by field linguists and computer scientists.

In this context, explorations in Automatic Speech Recognition on field data are ongoing. After an introduction to the Elpis project, fresh experiments will be presented: using Huggingface's Transformers, and Wav2Vec2.0 from Facebook AI. The goal is to allow an audience of (budding) linguists to better understand how the tools work, so as to address the challenges of interdisciplinary collaborations with computer scientists.

Résumé

Les laboratoires LLACAN et LACITO sont engagés dans des projets exploratoires qui visent à exploiter le potentiel des méthodes informatiques afin de faciliter les tâches de documentation des langues en danger. Les outils fondés sur l'apprentissage machine peuvent aider efficacement aux tâches d'annotation linguistique : transcription, glosage, traduction. Mais le traitement automatique reste peu utilisé, notamment parce que la technologie est encore nouvelle (et évolue rapidement), et qu'on manque d'interfaces simples et conviviales. Nos laboratoires ambitionnent une co-construction de modèles et d'outils par des linguistes de terrain et des informaticiens.

Dans ce cadre, des explorations en Reconnaissance Automatique de la Parole sur données de terrain sont en cours. Après une présentation globale du projet "Elpis", des expériences en cours seront présentées (qui recourent aux Transformers de Huggingface, et à Wav2Vec2.0 de Facebook AI). L'objectif est de permettre à un public de linguistes de mieux comprendre le fonctionnement des outils et les enjeux des collaborations interdisciplinaires avec des informaticiens.

Who am I?

Master 2 student in Natural Language Processing, University of Lorraine, Nancy.

Graduation internship @Lacito, CNRS under the supervision of Séverine Guillaume, Guillaume Wisniewski (LLF, CNRS) & Alexis Michaud.

→ Topics: Natural Language Processing (NLP), Artificial Intelligence (AI), Automatic Speech recognition (ASR).

What will this presentation talk about?

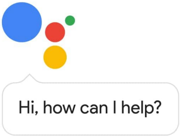
→ Automatic Speech Recognition tools: Kaldi, ESPnet, Wav2Vec2 (+Persephone).

→ Why and how it can be helpful for the documentation of low resource languages.

Outline

Introduction

Cortana

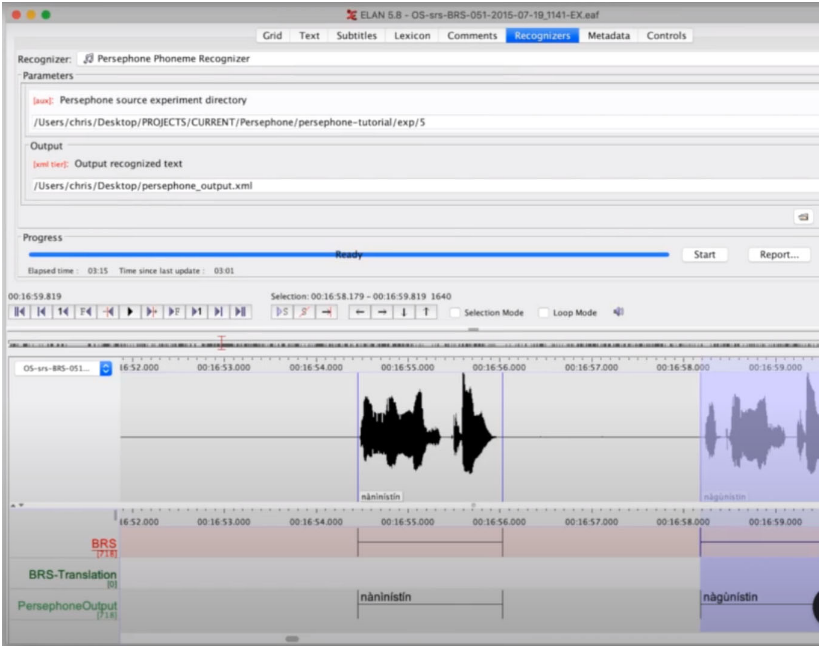


Google Assistant

How it began

Project history: narrated in a 2018 paper: “Integrating automatic transcription into the language documentation workflow: Experiments with Na data and the Persephone toolkit” [1]
https://halshs.archives-ouvertes.fr/halshs-01841979

- Na corpus (Pangloss Collection) used by an Australian PhD student
- promising tests training an acoustic model and transcribing fresh data
- efforts to set up a sustained collaboration
- creation of Elan plugin, Persephone-Elan (Chris Cox)
  https://www.youtube.com/watch?v=-P8OMMmP31E


ELAN 5.8 - OS-srs-BRS-051-2015-07-19_1141-EX.eaf
Grid
Text
Subtitles
Lexicon
Comments
Recognizers
Metadata
Controls
Recognizer: Persephone Phoneme Recognizer
Parameters
[aux]: Persephone source experiment directory
/Users/chris/Desktop/PROJECTS/CURRENT/Persephone/persephone-tutorial/exp/5
Output
[xml tier]: Output recognized text
/Users/chris/Desktop/persephone_output.xml
Progress
Ready
Start
Report...
Elapsed time : 03:15 Time since last update : 03:01
00:16:59.819
Selection: 00:16:58.179 - 00:16:59.819 1640
Selection Mode
Loop Mode
BRS
BRS-Translation
PersephoneOutput
nàninístín
nàgùnístín

2019: funding from ANR

2019: joined forces with a team based in Australia: Elpis ἐλπίς, 'hope'

2020: funding from the Institute for Linguistic Heritage and Diversity, ILARA-EPHE *https://ilara.hypotheses.org/*

2021: 6-month internship by Cécile Macaire, supervised by Séverine Guillaume and Guillaume Wisniewski

Automatic Speech Recognition (ASR)

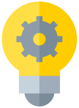
Automatic Speech Recognition (ASR): a simple definition

"Recognition and translation of spoken languages into text by computers." *[2]*

Interdisciplinary subfield of:


Computer science


Linguistics


Electrical engineering

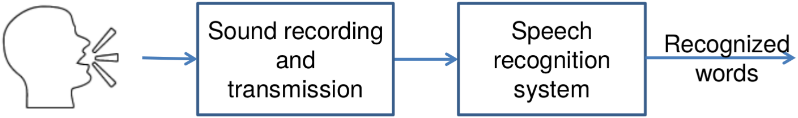


Figure 1: Simplified diagram of an automatic speech recognition system.

A brief history

In the seventies, rule-based approaches which:

- split the speech signal into fixed-size slices (about 10 ms),
- and tried to identify phonemes, words and sentences.

Since a few decades, data-based learning approaches:

- Acoustic templates: relies on comparing an unknown acoustic form to known reference templates using Dynamic Time Warping (DTW).
- Hidden Markov models (HMM): statistical models representing the pronunciation of a phoneme or of a word.
- **Neural Networks**: most efficient and most often used approach since c. 2010.

Why is it possible?
→ Large available corpora with labeled data, larger computational resources.

Neural networks (NN)

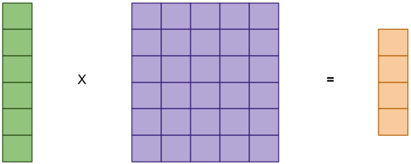
→ **Machine learning** — using statistics to enable machines to learn.

→ **Deep learning**

- artificial neural networks: inspired by human biology,
- algorithms capable of autonomous improvement through modelling,
- based on large amounts of data.

How does it work?

- Inputs are encoded into vectors.
- A NN is composed of multiple layers.
- Transformation of an input vector into an output vector by a single layer.
- The output vector is the multiplication of the input vector with a set of parameters (weights).



- How to find these parameters? By the deep learning system.

Measures of performance

- Substitution (S)

REF: Paris
HYP: Poris

- Insertion (I)

REF: I want to leave the country.
HYP: However I want to leave the country.

- Deletion (D)

REF: I want to leave the country.
HYP: I want leave the country.

Word Error Rate (WER)[1]: number of deletions, substitutions and insertions divided by the total number of correct words (C).

$$
\begin{aligned}
WER &= (S + D + I)/N \\
&= (S + D + I)/(S + D + C)
\end{aligned}
\tag{1}
$$

[1] `https://huggingface.co/metrics/wer`

Measures of performance (2)

- **Character Error Rate (CER)**: character level.

REF: I * w a n t * t o * e a t
HYP: I * w i l l * n o t * e a t * t h a t.

- **Phoneme Error Rate (PER)**: phoneme level.

→ THE LOWER THE BETTER

Do ASR systems work?

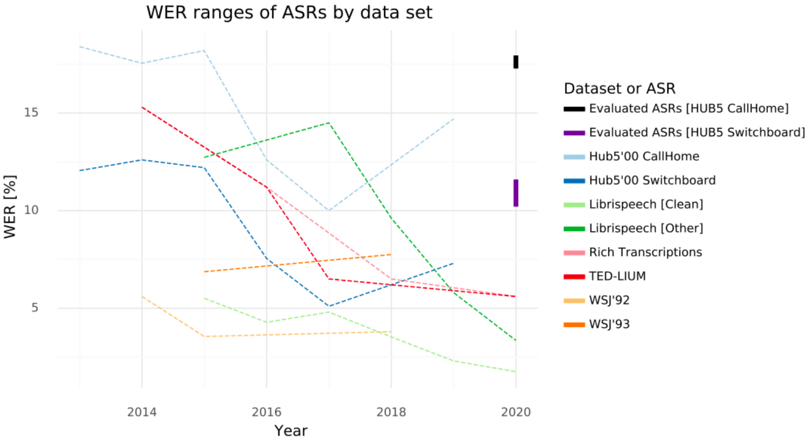


Figure 2: WER ranges in ASR systems published in the last 5 years with benchmark datasets [3].

ASR for low resource languages

→ Spectacular progress in automatic speech recognition over the past decade [4]–[6], and especially for low resource languages [7], [8].

→ A few hours of data (speech and annotations) are enough to learn systems capable of automatically recognizing phonemes with sufficient accuracy to meet the needs of field linguists [9], [10].

Elpis: A graphical interface for ASR

What is Elpis ?

Software developed by the Australian Research Council in the Centre of Excellence for the Dynamics of Language [2] with participation of Daan van Esch (Google), Benjamin Galliot (LACITO) and others

Graphical interface to:

- **train** your own acoustic model for speech recognition,
- and automatically **transcribe** speech recordings.

Two available speech recognition models:

Figure 3: Kaldi [5].

Figure 4: ESPnet [11].

[2] *http://www.dynamicsoflanguage.edu.au/*

How to install it?

Two possibilities:

1. Install with **Docker**,
2. Install with Google Cloud Platform.

Prerequisites:

Link to the documentation:
https://elpis.readthedocs.io/en/latest/.

1st speech recognition engine: Kaldi [12]

Open source toolkit for ASR.
→ **Hidden Markov models** + **Deep Neural Networks**.

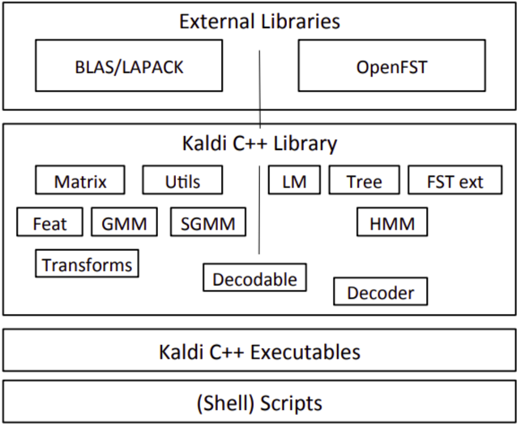


Figure 5: A simplified view of the different components of Kaldi.

2nd speech recognition engine: ESPnet [11]

End-to-End speech processing toolkit.

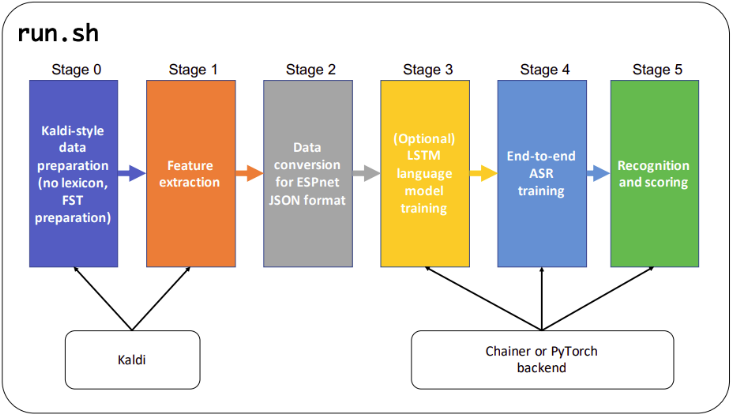
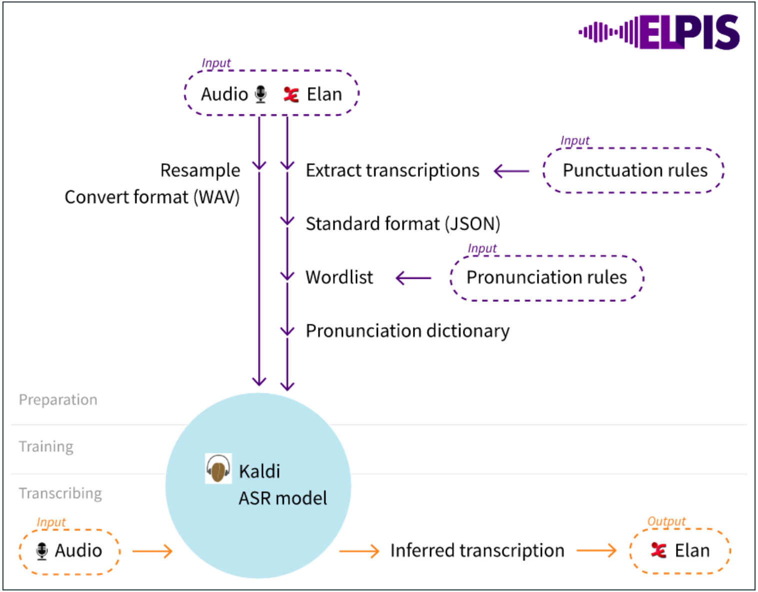


Figure 6: Experimental flow of standard ESPnet recipe.

Workflow



ELPIS
Input
Audio
Elan
Resample
Convert format (WAV)
Extract transcriptions
Input
Punctuation rules
Standard format (JSON)
Input
Pronunciation rules
Wordlist
Pronunciation dictionary
Preparation
Training
Transcribing
Kaldi
ASR model
Input
Audio
Inferred transcription
Output
Elan

Input files

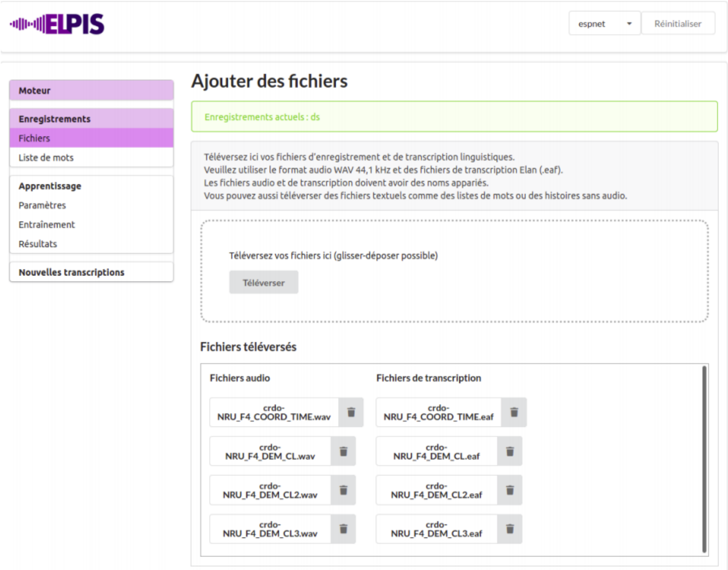
Pairs of audio (in WAV) and transcriptions (in .eaf format).

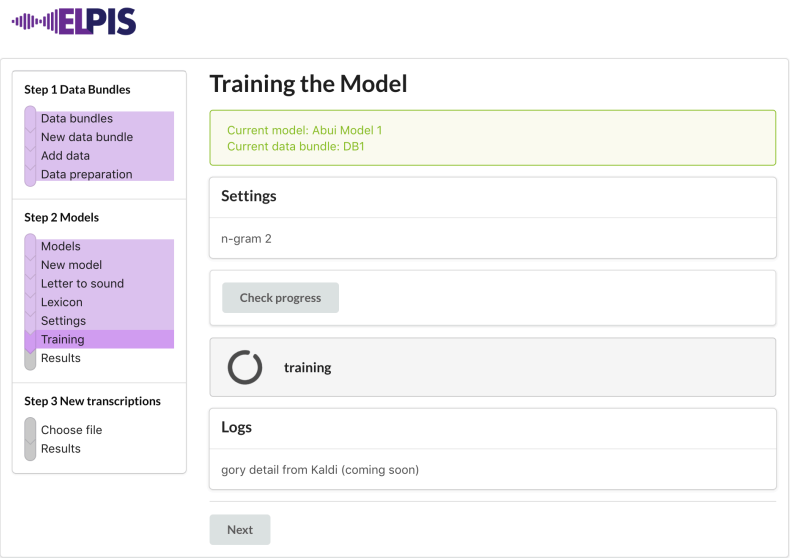
EAF: special XML based format called ELAN Annotation Format.


```xml
<?xml version="1.0" encoding="UTF-8"?>
<ANNOTATION_DOCUMENT AUTHOR="" DATE="2017-07-06T13:35:23+10:00" FORMAT="2.8" VERSION="2.8"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="http://www.mpi.nl/tools/elan/EAFv2.8.xsd">
    <HEADER MEDIA_FILE="" TIME_UNITS="milliseconds">
        <MEDIA_DESCRIPTOR MEDIA_URL="file:///Users/neinheim/Documents/GitHub/asr-
daan/toy_corpus/data/1_1_1.wav" MIME_TYPE="audio/x-wav" RELATIVE_MEDIA_URL="./1_1_1.wav"/>
        <PROPERTY NAME="URN">urn:nl-mpi-tools-elan-eaf:a680ab14-1485-4eda-a9fb-0e0003430d2f</PROPERTY>
        <PROPERTY NAME="lastUsedAnnotationId">1</PROPERTY>
    </HEADER>
    <TIME_ORDER>
        <TIME_SLOT TIME_SLOT_ID="ts1" TIME_VALUE="290"/>
        <TIME_SLOT TIME_SLOT_ID="ts2" TIME_VALUE="1910"/>
    </TIME_ORDER>
    <TIER LINGUISTIC_TYPE_REF="default-lt" PARTICIPANT="SL" TIER_ID="Phrase">
        <ANNOTATION>
            <ALIGNABLE_ANNOTATION ANNOTATION_ID="a1" TIME_SLOT_REF1="ts1" TIME_SLOT_REF2="ts2">
                <ANNOTATION_VALUE>amakaang di kaai hada muila</ANNOTATION_VALUE>
            </ALIGNABLE_ANNOTATION>
        </ANNOTATION>
    </TIER>
    <LINGUISTIC_TYPE GRAPHIC_REFERENCES="false" LINGUISTIC_TYPE_ID="default-lt" TIME_ALIGNABLE="true"/>
    <CONSTRAINT DESCRIPTION="Time subdivision of parent annotation's time interval, no time gaps allowed within
this interval" STEREOTYPE="Time_Subdivision"/>
    <CONSTRAINT DESCRIPTION="Symbolic subdivision of a parent annotation. Annotations refering to the same
parent are ordered" STEREOTYPE="Symbolic_Subdivision"/>
    <CONSTRAINT DESCRIPTION="1-1 association with a parent annotation" STEREOTYPE="Symbolic_Association"/>
    <CONSTRAINT DESCRIPTION="Time alignable annotations within the parent annotation's time interval, gaps are
allowed" STEREOTYPE="Included_In"/>
</ANNOTATION_DOCUMENT>
```

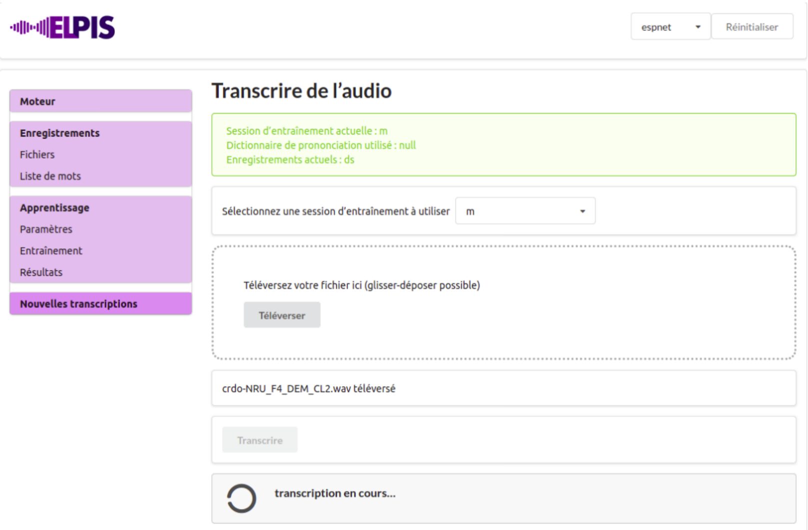

Figure 7: Example of an .eaf annotation file.

Uploading the files




ELPIS
Step 1 Data Bundles
Data bundles
New data bundle
Add data
Data preparation
Step 2 Models
Models
New model
Letter to sound
Lexicon
Settings
Training
Results
Step 3 New transcriptions
Choose file
Results
Training the Model
Current model: Abui Model 1
Current data bundle: DB1
Settings
n-gram 2
Check progress
training
Logs
gory detail from Kaldi (coming soon)
Next

Transcription of a new file



Results on 5 corpora

*Character Error Rate on 5 corpora with the **ESPnet recipe**.*
→ Corpora available in the Pangloss Collection [3].

Langue	Nb locuteurs	Type	Taille (mn)	CER (%)
Na	1	*Récits spontanés*	273	14.5
Na	1	*Expressions élicitées*	188	4.7
Chatino	1	*Parole lue*	81	23.5
Japhug	1	*Récits spontanés*	170	12.8
Bashkir	36	*Récits spontanés*	273	33

Figure 8: Information on the evaluation datasets used and the character error rate performance of the current recipe.

[3] https://pangloss.cnrs.fr/

How many training data to train such models?

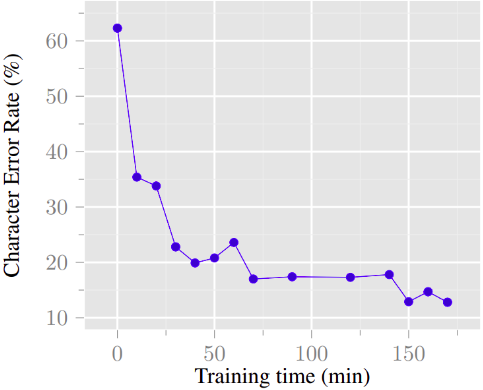


Figure 9: Character error rate for Japhug with different amount of training data, using the ESPnet recipe included in Elpis.

Remark: up to you to define which CER to achieve.

Elpis: A quick recap

- User-friendly interface.
- Two available engines: Kaldi & ESPnet.
- Competitive CER results on low resource languages.

To keep in mind:

Good quality of speech will improve the performances.

A transcription has to be consistent with the audio file.

What's next? Wav2Vec2, a new automatic speech recognition model.

Wav2Vec2

Wav2vec2 [13] by Facebook AI

Automatic Speech Recognition model by Facebook AI [13].

Available in the Transformers library v4.3.0[4] by HuggingFace[5].

Why this choice ?

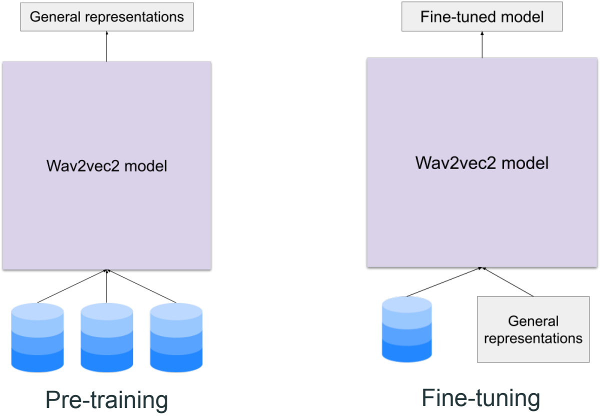
Competitive results compared to the most advanced ASR systems with a pre-training step, followed by a fine-tuning on labelled speech data.

[4] *https://huggingface.co/transformers/*

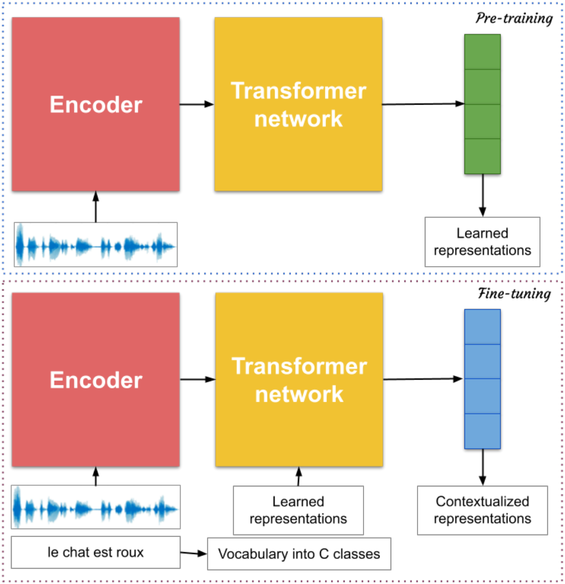
[5] *https://huggingface.co/*

Wav2Vec2: Self-supervised training

1. **Pre-training**: pre-train a wav2vec 2.0 model on the unlabeled data (using self-supervised learning approach).
2. **Fine-tuning**: fine-tuning these representations learned during the pre-training on labelled data.



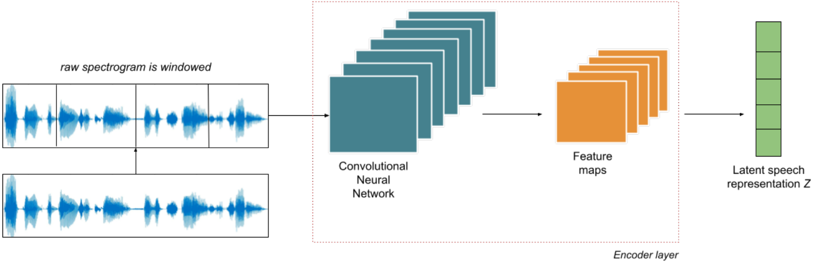
Model Architecture



Feature Encoder

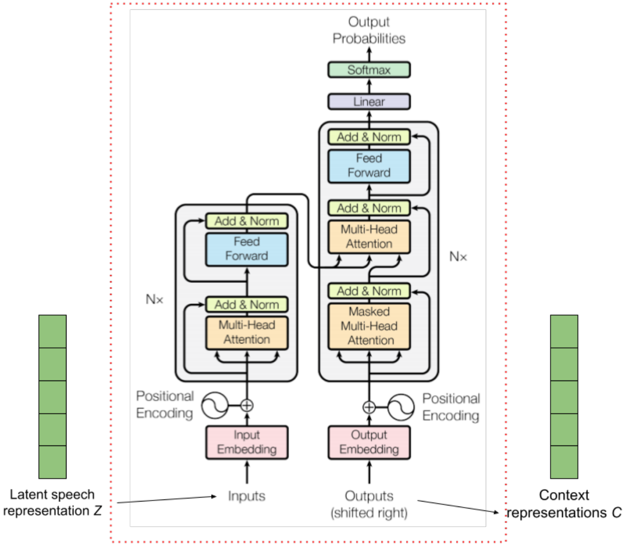
Feature encoder:

- builds a **latent speech representation**.
- each speech frame is represented by a vector.



Quantization module: we discretize the output of the feature encoder z to a finite set of speech representations via product quantization.

Transformer model [6] [14]



Transformer architecture (context network)

[6] https://ledatascientist.com/a-la-decouverte-du-transformer/

Self-attention

Attention: measure the extent to which two elements of two sequences are linked.

$\rightarrow$ Self-attention: the interdependence of the different speech utterances of the same sequence in order to associate a relevant representation (encoding) to it.

To build the representation of a speech:

- “look” at all the other speech utterances of the speech file,
- and adapt the latent representation accordingly.

Self-attention: Visualization

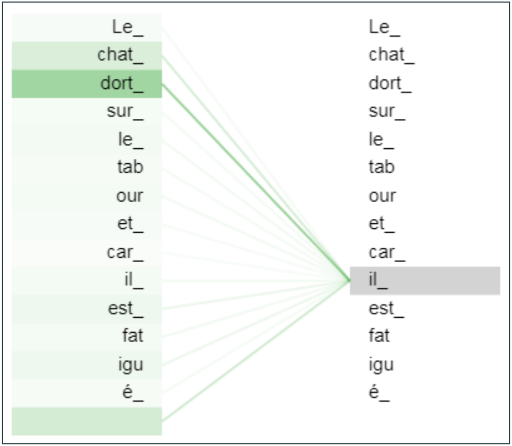


Figure 10: Visualisation of a self-attention layer.

Decoder

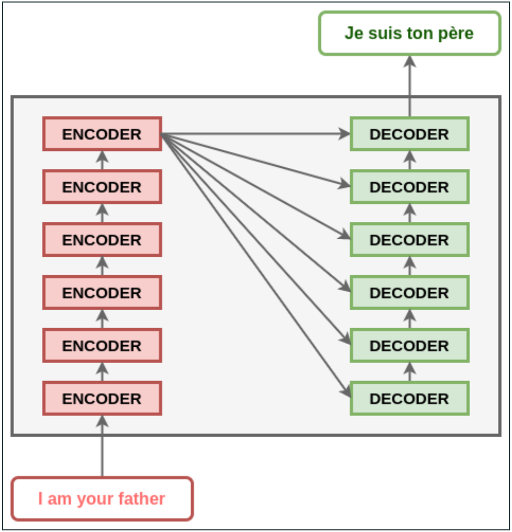
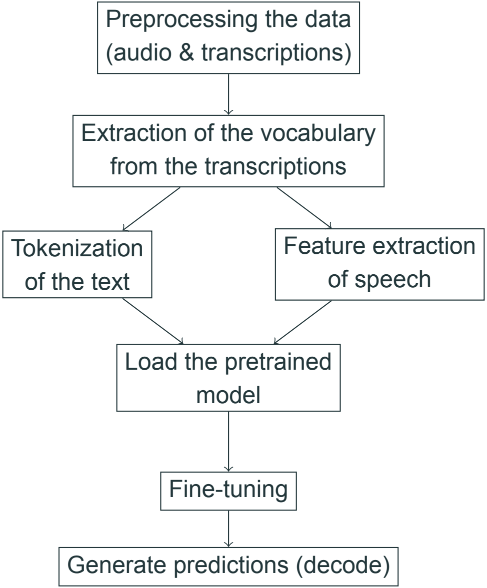


Figure 11: Transformer Architecture.

Pipeline to fine-tune your own data



Corpora : Na & Japhug

Corpus	Yongning Na	Japhug
Number of files	57 <audio, xml>	357 <audio, xml>
Number of sentences	2484	31864
Total duration (in minutes)	209.52 ($\approx$ 3h30)	1907.57 ($\approx$ 31h47)
Number of speakers	1 female speaker	2 male and 2 female speakers

Table 1: Information and statistics of the Na and the Japhug corpora.

Example of a transcription file

```xml
<?xml version="1.0" ?>
<!DOCTYPE TEXT SYSTEM "https://cocoon.huma-num.fr/schemas/Archive.dtd">
<TEXT id="crdo-JYA_HIST140512_ALIBABA" xml:lang="jya">
        <HEADER>
                <TITLE>crdo-JYA_HIST140512_ALIBABA</TITLE>
                <SOUNDFILE href="hist140512_alibaba.wav"/>
        </HEADER>
        <S id="S001">
                <AUDIO start="0.28" end="2.99"/>
                <FORM kindOf="phono">ʑa-tɕi ɕi? tɤ-zgra mɯ́j-ʁdɯɣ? </FORM>
        </S>
        <S id="S002">
                <AUDIO start="6.81" end="11.0"/>
                <FORM kindOf="phono">kukutɕu, kɯɕɯŋgɯ tɤ-tɕɯ alibaba kɤ-ti ci pjɤ-tu tɕe,</FORM>
        </S>
        <S id="S003">
                <AUDIO start="11.72" end="15.01"/>
                <FORM kindOf="phono">nɯ chondɤre iɕqha, tɕaxpa kɯβdɤsqi kɤ-ti ɲɯ-ŋu.</FORM>
        </S>
        <S id="S004">
                <AUDIO start="15.93" end="22.64"/>
                <FORM kindOf="phono">tɕendɤre, iɕqha kɯɕɯŋgɯ tɕe nɤkinɯ, iɕqha nɯ,</FORM>
        </S>
```

Figure 12: Sample of an XML transcription file entitled *140512_alibaba*.

Preprocessing the data

Data preprocessing :

- Extraction of each sentence and the corresponding audio segment from the file.
- Split the data into training, validation and test sets, with a ratio of 70, 15 and 15 % respectively.

Language	Na	Japhug
Training	2110 sentences ($\approx$ 3 hours)	3000 sentences ($\approx$ 3h30)
Test	374 sentences	350 sentences
Training time	$\approx$ 4h30	$\approx$ 14h

- Cleaning of the transcriptions (deletion or substitution of specific characters (punctuation, etc.) and audio file conversion (WAV format in mono, 16kHz sampling rate).

Preprocessing the data (2)

Reference	Processed
ʑo kɤ-ndza rga-nɯ	ʑo kɤndza rganɯ
nɯ... praʁɕku kɤti tu,	nɯ praʁɕku kɤti tu
rɟɤlpu nɯ kɯ li ”mɤʑɯ	rɟɤlpu nɯ kɯ li mɤʑɯ
/kɯnɯ/ iɕqha,	kɯnɯ iɕqha

Table 2: Extract of sentences before and after the preprocessing step on Japhug data.

Extraction of the vocabulary + Tokenization

- Generation of a dictionary with a unique ID for each token (vocabulary):

  ”Z”: 0, ”i”: 1, ”h”: 2, ”s”: 3, ”ɟ”: 4, ” ”: 5, ”p”: 6, ”l”: 7, ”ŋ”: 8, ”ɢ”: 9, ”β”: 10, ”x”: 11, ”ə”: 12, ”y”: 13, ”c”: 14, ”a”: 15, ”o”: 16, ”ɣ”: 17, ”χ”: 18, ”f”: 19, ”ʑ”: 20, ”q”: 21, ”k”: 22, ”|”: 23, ...

- Tokenization: sentences cut into characters, and transform into a list of integer.

  Sentence: tɕeri tɤmda ʑo tɕendɤre
  Tokenization: [48, 24, 43, 36, 1, 5, 48, 42, 38, 45, 15, 5, 20, 16, ...]

Chosen pretrained model

→ wav2vec2-XLSR-53 pre-trained on 53 languages (**multilingual**) (*facebook/wav2vec2-large-xlsr-53*)[7].

3 datasets:

- MLS: Multilingual LibriSpeech (**8 languages, 50.7k hours**): Dutch, English, French, etc.
- CommonVoice (**36 languages, 3.6k hours**): Arabic, Basque, Breton, Chinese (CN), Chinese (HK), Chinese (TW), Chuvash, Dhivehi, Dutch, English, Esperanto, Latvian, Mongolian, Persian, Portuguese, Welsh, etc.
- Babel (**17 languages, 1.7k hours**): Assamese, Bengali, Cantonese, Cebuano, Georgian, Haitian, etc.

→ **The model can be used on any languages**.

[7] https://huggingface.co/models?search=facebook/wav2vec2

Results

Ref:	õ˧˥ tʰv̩˧qo˧dʑo˧ kʰwɤ˧pʰv̩˧qo˧gɤ˧ pi˧hĩ˧ ʈʂʰɯ˧ne˧ʝi˥ tʰi˧di˩kv̩˩tsɯ˩ ʈʂʰɯ˧ne˧ ʝi˥zo˩ gɤ˩ʑi˥ tʰi˧di˩kv̩˩tsɯ˩
Hyp:	õ tʰv̩˧qo˧dʑo˧ kʰwɤ˧pv̩˧qo˥ gɤ˧ pi˧ ʈʂʰɯ˧ne˧ʝi˥ tʰi˧di˩kv̩˩tsɯ˩ ʈʂʰɯ˧ne˧ʝi˥zo˩ gɤ˩ʑi˥ tʰi˧di˩kv̩˩tsɯ˩
Ref:	njæ˧ɻ̩˩ ə˧ʝi˧ʂɯ˥ʝi˩ mo˧ ɖʐɤ˥bi˩dʑo˩ mo˧kɤ˩dzi˩ʈʰæ˩ko˩ pi˩zo˩ no˧ mo˧kɤ˩si˩dzi˩ ə˧sɯ˥
Hyp:	njæ˧ɻ̩˩ ə˧ʝi˧ʂɯ˥ʝi˩ mo˧ɖʐɤ˥bi˩dʑo˩ mo˧kɤ˩dʑɯ˩ʈʰæ˩qo˩ pi˩zo˩ mv̩˧mo˧kɤ˩se˩dzi˩ ə˧sɯ˥
Ref:	tʰi˧tɕɤ˩dʑo˩ tʰi˩˥ ɖɯ˧pʰo˧hĩ˧bv̩˧ ʝi˧qv̩˥ ɖɯ˧ɭɯ˧ mɤ˧dʑo˧zo˧
Hyp:	tʰi˧tɕɤ˩ dʑo˩ tʰi˩˥ ɖɯ˧pʰo˧hĩ˧bv̩˧ ʝi˧qv̩˥ ɖɯ˧ɭɯ˧ mɤ˧dʑo˧zo˩
Ref:	njæ˧sɯ˩kv̩˩ ə˧ʝi˧ʂɯ˥ʝi˩dʑo˩ zo˩no˥ njæ˧sɯ˩kv̩˩ tʰi˧dzi˩hĩ˩ tʰv̩˩ʁwɤ˩dʑo˩ zo˩no˥ ɖæ˩mi˧ ə˧lɑ˧ʁwɤ˧ pi˧kv̩˩tsɯ˩ mv̩˩ ə˧ʝi˧ʂɯ˥ʝi˩
Hyp:	njæ˧sɯ˩kv̩˩ ə˧ʝi˧ʂɯ˥ʝi˩dʑo˩ zo˩no˥ njæ˧sɯ˩kv̩˩ tʰi˧dzi˩hĩ˩tʰv̩˩ʁwɤ˧ dʑo˩ zo˩no˥ ɖæ˩mi˧ ə˧lɑ˧ʁwɤ˧ pi˧kv̩˩tsɯ˩ mv̩˩ ə˧ʝi˧ʂɯ˥ʝi˩
Ref:	tʰi˩˥ mɤ˧ɖwæ˩ mɤ˧ɖwæ˩ njɤ˧ ə˧v̩˧˥ ɖɯ˩hĩ˩ɳɯ˥ wɤ˩˥ ə˧tso˧ ɖwæ˩tso˩ dʑo˩ njɤ˧ɳɯ˧ wɤ˩˥ hwæ˧bi˧ pi˧
Hyp:	tʰi˩˥ mɤ˧ɖwæ˩ mɤ˧ɖwæ˩ njɤ˧ ə˧v̩˧˥ ɖɯ˩hĩ˩ɳɯ˩ wɤ˩˥ ə˧tso˧ ɖwæ˩tso˩dʑo˩ njɤ˧ɳɯ˧ wɤ˩˥ hwæ˧bi˧ pi˧
Ref:	jɤ˧ŋɤ˧ɳɯ˧dʑo˥ əəə… ho˧di˧ po˧hɯ˥
Hyp:	jɤ˧ŋɤ˧ɳɯ˧dʑo˥ əə… ho˧di˧ po˧hɯ˥
Ref:	əəə… zo˩no˥ si˧kʰɯ˧ɭɯ˧bv̩˥pi˩zo˩ zo˩no˧dʑo˥ tsʰo˧jo˩ mv̩˩ʈʂæ˩ɲi˩mæ˩ ə˩gi˩
Hyp:	əə… zo˩no˥ si˧kʰɯ˧ɭɯ˧bv̩˥ pi˧zo˩ zo˩no˧dʑo˥ tsʰo˧ʝi˧ mv̩˩ ʈʂæ˩ɲi˩ze˩mæ˩ ə˩gi˩
Ref:	tʰi˩˥ gi˩˥ dʑo˩ mɤ˧ɲi˩ze˩ õ˧˥ hɤ˩ ʈʂɤ˩tso˥ mɤ˧ɲi˩
Hyp:	tʰi˩˥ gi˩˥ dʑo˩ mɤ˧ɲi˩ze˩ õ˧˥ hɤ˩ʈʂɤ˩tso˧ mɤ˧ɲi˩
Ref:	tʰi˩˥ tʰv̩˧ɲi˧ʑ ɲi˧ʑi˩ ɲi˩tsɯ˩ mv̩˩ ɖɯ˧ʑi˩ɻ̩˩dʑo˩ zo˧ ɖɯ˧v̩˧ tɕʰi˧ pi˧zo˩
Hyp:	tʰi˩˥ ʈʂʰɯ˧ɲi˧ʑi˩ ɲi˧ʑi˩ ɲi˩tsɯ˩ mv̩˩ ɖɯ˧ʑi˩ɻ̩˩dʑo˩ zo˧ ɖɯ˧v̩˧ tɕʰi˧bi˥ pi˧zo˩
Ref:	tʰi˩˥ tsʰi˧qʰæ˧˥ dʑo˩ wɤ˩˥ ʈʂo˩bo˩ ti˥ mmm… ʈʂwæ˧ʑi˧qʰwɤ˧ gv̩˩ɲi˩ pi˩ ŋwɤ˧ʈʂwæ˧ ʑi˧qʰwɤ˧ gv̩˩
Hyp:	tʰi˩˥ tsʰi˧qʰæ˧˥ dʑo˩ wɤ˩˥ ʈʂo˩bo˩ ti˥ mm… ʈʂwæ˧ ʑi˧qʰwɤ˧ gv̩˩ɲi˩ pi˩ ŋwɤ˧ʈʂwæ˧ ʑi˧qʰwɤ˧ gv̩˩
Ref:	njæ˧sɯ˩kv̩˩ qo˩qɑ˩˥ ɑ˩ʁo˧ dʑo˩ pi˧dʑo˩ ɖʐɯ˧qo˩ni˩hĩ˩ɳɯ˩dʑo˩ əəə… ʐwæ˧ tʰi˧se˥ le˧mv̩˥kv̩˩ze˩
Hyp:	njæ˧sɯ˩kv̩˩ qo˩qɑ˩˥ ɑ˩ʁo˧ dʑo˩ pi˩dʑo˩ ɖʐɯ˧qo˩ni˩hĩ˩ɳɯ˩dʑo˩ ə… ʐwæ˧ tʰi˧se˥ le˧mv̩˥kv̩˩ze˩

How many training data ?

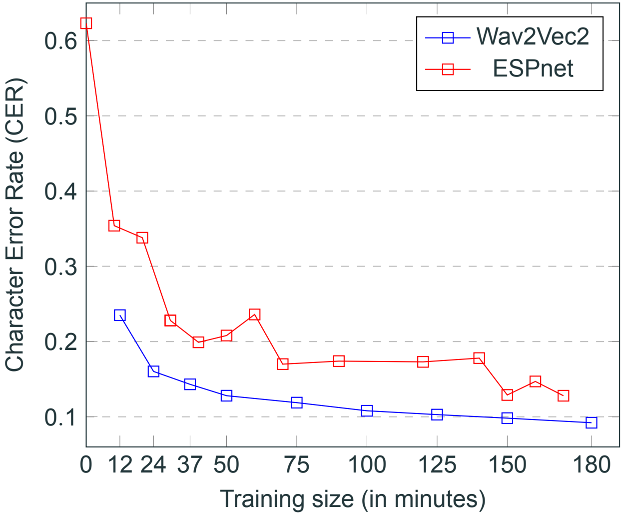


Figure 13: CER score with respect to different training sizes (in minutes).

Can the decoding be improved?

→ **CTC decoder** [8]: makes a conditional independence assumption over the characters in the output sequence at each timestep *t*.

By default, use of the greedy search algorithm defined as:

$$w_t = argmax_w P(w|w_{1:t-1}) \quad (2)$$

where w_t is the token probability at timestep *t*.

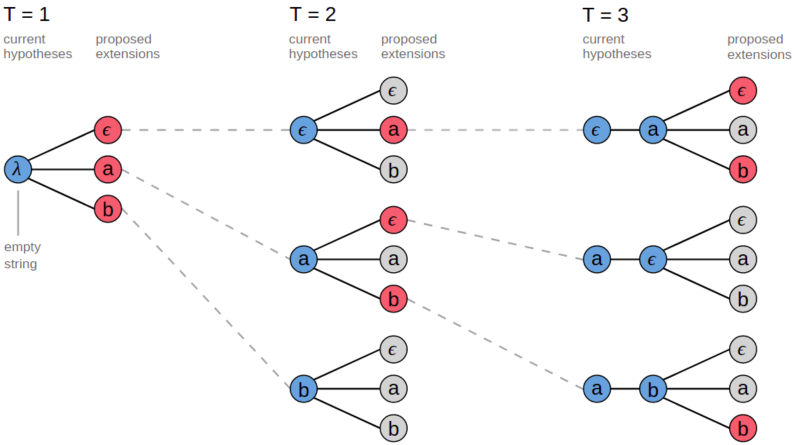
t

w	t_1	t_2	t_3	t_4	t_5	t_6
a	0.2	0.1	0.45	0.6	0.02	0.2
b	0.1	0.9	0.62	0.05	0.12	0.98
c	0.43	0.06	0.08	0.56	0.9	0.53
d	0.33	0.5	0.07	0.18	0.32	0.24
e	0.76	0.96	0.7	0.61	0.04	0.6

[8] https://distill.pub/2017/ctc/

Can the decoding be improved? A first solution

Beam search algorithm: keeps a fixed number of beams hypotheses at each time step.



A standard beam search algorithm with an alphabet of $\{\epsilon, a, b\}$ and a beam size of three.

Can the decoding be improved? A first solution (2)

Top k hypotheses by the beam search algorithm:

K	Oracle CER
50	7.05 % (- 0.92 %)
100	6.88 % (- 1.09 %)
150	6.78 % (- 1.19 %)
200	6.71 % (- 1.26 %)
250	6.67 % (- 1.3 %)

Table 3: Oracle CER scores on the top-k hypotheses for the Na.

Can the decoding be improved? A second solution

Language model: used to know the probability of sentences.

Example: ”I love Computer Science”

2-grams: P(love | I), P(Computer | love)
3-grams: P(Computer | I love)

Word-based language model:

N-gram KenLM	2	3
CER	9.27 % (+ 1.3 %)	9.25 % (+ 1.28 %)

Table 4: CER scores with different n-gram KenLM language model on the Na.

N-gram KenLM	2	3
CER	9.17 % (+ 0.3 %)	9.12 % (+ 0.25 %)

Table 5: CER scores with different n-gram KenLM language model on the Japhug.

Limitations

On the data:

- High quality data.
- Minimum 1 hour of labeled data.
- Required processing step.

During the training:

- High computational resources (GPU).
- Several hours / days to train a model.

On the performances:

- Poor performances on a recording with a speaker not present in the training corpus.
- Poor performances when several languages are present in one recording.

Conclusion

- The **field is evolving rapidly** - in one decade, performances reached less than 5 % of errors.
- **Powerful ASR systems** to work with low resource languages.
  $\rightarrow$ Kaldi / ESPnet / Wav2Vec2
- But, still some **limitations** are encountered (resources, computational, biases).

Thank you for your attention.

Any questions?

References

[1] A. Michaud, O. Adams, T. Cohn, G. Neubig, and S. Guillaume, "Integrating automatic transcription into the language documentation workflow: Experiments with Na data and the Persephone toolkit," *Language Documentation and Conservation*, vol. 12, pp. 393–429, 2018, Dans HAL: https://halshs.archives-ouvertes.fr/halshs-01841979/. [Online]. Available: `http://hdl.handle.net/10125/24793`.

[2] Wikipedia, *Speech recognition*, `https://en.wikipedia.org/wiki/Speech_recognition`, Online; accessed 1st July 2021, 2021.

[3] P. Szymański, P. Żelasko, M. Morzy, A. Szymczak, M. Żyła-Hoppe, J. Banaszczak, L. Augustyniak, J. Mizgajski, and Y. Carmiel, "Wer we are and wer we think we are," *arXiv preprint arXiv:2010.03432*, 2020.

[4] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, *et al.*, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal processing magazine*, vol. 29, no. 6, pp. 82–97, 2012.

[5] M. Ravanelli, T. Parcollet, and Y. Bengio, "The pytorch-kaldi speech recognition toolkit," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2019, pp. 6465–6469.

[6] W. Zhou, W. Michel, K. Irie, M. Kitza, R. Schlüter, and H. Ney, "The rwth asr system for ted-lium release 2: Improving hybrid hmm with specaugment," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2020, pp. 7839–7843.

[7] L. Besacier, E. Barnard, A. Karpov, and T. Schultz, "Automatic speech recognition for under-resourced languages: A survey," *Speech communication*, vol. 56, pp. 85–100, 2014.

[8] D. van Esch, B. Foley, and N. San, “Future directions in technological support for language documentation,” in *Proceedings of the Workshop on Computational Methods for Endangered Languages*, vol. 1, 2019.

[9] A. Michaud, O. Adams, C. Cox, and S. Guillaume, “Phonetic lessons from automatic phonemic transcription: Preliminary reflections on na (sino-tibetan) and tsuut’ina (dene) data,” in *ICPhS XIX (19th International Congress of Phonetic Sciences)*, 2019.

[10] G. Wisniewski, A. Michaud, and S. Guillaume, “Phonemic transcription of low-resource languages: To what extent can preprocessing be automated?” In *1st Joint SLTU (Spoken Language Technologies for Under-resourced languages) and CCURL (Collaboration and Computing for Under-Resourced Languages) Workshop*, European Language Resources Association (ELRA), 2020, pp. 306–315.

[11] S. Watanabe, T. Hori, S. Karita, T. Hayashi, J. Nishitoba, Y. Unno, N. E. Y. Soplin, J. Heymann, M. Wiesner, N. Chen, *et al.*, “Espnet: End-to-end speech processing toolkit,” *arXiv preprint arXiv:1804.00015*, 2018.

[12] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, *et al.*, "The kaldi speech recognition toolkit," in *IEEE 2011 workshop on automatic speech recognition and understanding*, IEEE Signal Processing Society, 2011.

[13] A. Baevski, H. Zhou, A. Mohamed, and M. Auli, "Wav2vec 2.0: A framework for self-supervised learning of speech representations," *arXiv preprint arXiv:2006.11477*, 2020.

[14] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," *arXiv preprint arXiv:1706.03762*, 2017.