

Sarika

Khmer Text-to-Speech

Technology

Seanghay Yath

Senior ML Engineer @ DGC

`seanghay.dev@gmail.com`

Introduction

Currently, I am a Senior ML Engineer at the Digital Government Committee. I am also a Product Developer in my free time. I built tools like [KhmerDict.com](https://khmerdict.com) [KhmerScan.com](https://khmerscan.com) and more.

My interests are product design and product development and some deep learning.



Disclaimer

The views and opinions expressed herein are solely my own and do not reflect, represent, or constitute the official position, policy, or endorsement of any organization, company, institution, or employer with which I am or have been affiliated.

What is Text to Speech

A technology that converts written text into spoken audio. It's a core component of speech synthesis systems that enables computers to "read aloud" digital text in a human-like voice.



Why do we need it?



Content/News Creation



Visual Impairment Support



Virtual Assistants /
Robotics



Navigation Systems



Video Narration



Public Announcements



Audiobook Production



Customer Service

Human vs. TTS

Rich emotional expression and natural intonation	Consistent quality every time
Perfect contextual understanding	Available 24/7 instantly
Variable quality (fatigue, mood, health affects performance)	Can struggle with complex pronunciations
High cost and time investment	Low cost after initial setup
Difficult to modify after recording	Perfect for technical documentation and accessibility
Limited availability and scalability	Unlimited scalability

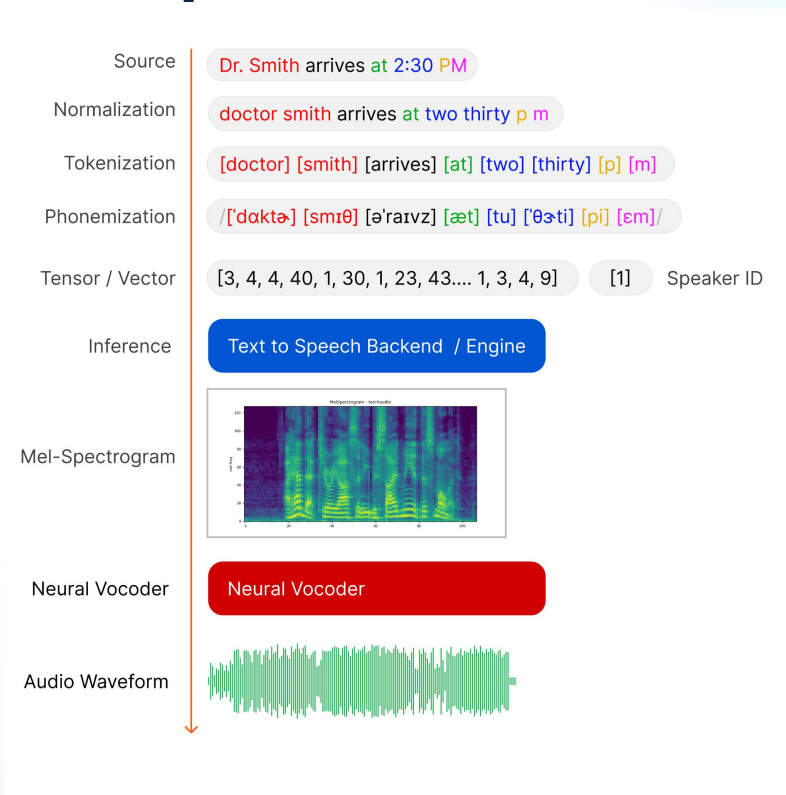
Popular TTS



ElevenLabs



Text-to-Speech Pipeline



Khmer Language

“Khmer has been influenced considerably by Sanskrit and Pali especially in the royal and religious registers, through Hinduism and Buddhism, due to Old Khmer being the language of the historical empires of Chenla and Angkor.” – Wikipedia

Khmer Language Challenge

- ❑ Khmer Character Encoding
- ❑ Khmer Normalization and Verbalization
- ❑ Khmer Word Boundary Tokenization
- ❑ Khmer Grapheme to Phoneme (Linguistic Processing)
- ❑ Khmer Pronunciation Toolkit (Linguistic Processing)

Khmer Encoding Normalization

ស៊ី

ស ៊ ី
179f 17bb 17b8

Normalized into

ស ី ី
179f 17ca 17b8

ស្តី

ស ៊ រ ៊ ត ី
179f 17d2 179a 17d2 178f 17b8

Normalized into

ស ៊ ត ៊ រ ី
179f 17d2 178f 17d2 179a 17b8

ចំប៉ី

ច ំ ប ៊ ី
1785 17c6 1794 17bb 17b8

Normalized into

ច ំ ប ៊ ី
1785 17c6 1794 17c9 17b8

លើក

ល ែ ី ក
179b 17c1 17b8 1780

Normalized into

ល ី ក
179b 17be 1780

ស៊ីប

ស ែ ៊ ី ប
179f 17c1 17bb 17b8 1794

Normalized into

ស ី ី ប
179f 17ca 17be 1794

នំប៉ង

ន ំ ប ៊ ៊ ង
1793 17c6 1794 17d0 17bb 1784

Normalized into

ន ំ ប ៊ ៊ ង
1793 17c6 1794 17c9 17d0 1784

<https://ភាសាខ្មែរ.com>

By **Marc Durdin** and **Makara Sok**

<https://animate-khmer-normalizer.vercel.app>

Khmer Normalizer Animate

Learn how Khmer normalizer works with visual animation. Stop typing to see the reordering animation.

```
npm install khmer-normalizer
```

<https://github.com/sillsdev/khmer-normalizer>

បង្កើត

Source Order:

ប ង ែ ី ៊ ក ត
1794 1784 17c1 17b8 17d2 1780 178f

Discrepancies in Khmer Unicode Character Ordering Rules and a Proposed Solution

<https://www.youtube.com/watch?v=mD-nrfvWtgc>

13 ways to encode ស៊ីប /səp/ 'to detect'



ស ី ើ ប				ស៊ីប	*	ស ុ េ ី ប				ស៊ីប	*	ស េ ី ុ ប				ស៊ីប	
9F	CA	BE	94			9F	BB	C1	B8			94	9F	C1	B8		BB
ស ី េ ី ប				ស៊ីប	*	ស ុ ី េ ប				ស៊ីប	*	ស ៉ េ ី ប				ស៊ីប	
9F	CA	C1	B8			94	9F	BB	B8			C1	94	9F	C9		C1
*	ស ី ី េ ប				ស៊ីប	*	ស ី ុ េ ប				ស៊ីប	*	ស ៉ ី េ ប				ស៊ីប
	9F	CA	B8	C1			94	9F	B8	BB			C1	94	9F	C9	
*	ស ុ ើ ប				ស៊ីប	*	ស ី េ ុ ប				ស៊ីប						
	9F	BB	BE	94			9F	B8	C1	BB		94					
*	ស ើ ុ ប				ស៊ីប	ស េ ុ ី ប				ស៊ីប							
	9F	BE	BB	94		9F	C1	BB	B8		94						

Word Tokenization

Khmer is a language that has no clear word boundary.

For example, the sentence: “អ្នកស្រី កែវ ចន្ទ បេះទុរេនលក់អស់៣០០តោនរួចហើយ ជាមួយតម្លៃ១៥០០០រៀលក្នុងមួយគីឡូក្រាម” → ['អ្នកស្រី', ' ', 'កែវ', ' ', 'ចន្ទ', ' ', 'បេះ', 'ទុរេន', 'លក់', 'អស់', '៣០០', 'តោន', 'រួចហើយ', ' ', 'ជាមួយ', 'តម្លៃ', '១៥០០០', 'រៀល', 'ក្នុង', 'មួយ', 'គីឡូក្រាម']

We can achieve this via khmercut Python library. Once it's broken down into piece, we can then feed it to the text processor.

Text Normalization / Verbalization

123.324 -> មួយរយ_ម្ភៃបី_ចុច_បីរយ_ម្ភៃបួន

123.001 -> មួយរយ_ម្ភៃបី_ចុច_សូន្យ_សូន្យ_មួយ

010123123 -> 0_10_12_31_23

1A 4444 -> 1 A កាវ៉េ4

Text Normalization / Verbalization

 Tha (ថា) - A Khmer Text Normalization and Verbalization Toolkit

<https://github.com/seanghay/tha>

```
pip install tha
```

```
## Number - Cardinals
assert tha.cardinals.processor("1234") == "មួយពាន់ពីររយសាមសិបបួន"
assert tha.cardinals.processor("1") == "មួយ"
assert tha.cardinals.processor("1_2") == "មួយពីរ"
assert tha.cardinals.processor("-1") == "ដកមួយ"
assert tha.cardinals.processor("10") == "ដប់"
assert tha.cardinals.processor("15") == "ដប់ប្រាំ"
assert tha.cardinals.processor("100") == "មួយរយ"
assert tha.cardinals.processor("10000") == "មួយម៉ឺន"
assert tha.cardinals.processor("10000.234") == "មួយម៉ឺនពីររយសាមសិបបួន"
assert tha.cardinals.processor("-10000.234") == "ដកមួយម៉ឺនពីររយសាមសិបបួន"
assert tha.cardinals.processor("-10000,234") == "ដកមួយម៉ឺនពីររយសាមសិបបួន"

## Number - Decimals
assert tha.decimals.processor("123.324") == "មួយរយវិញ្ញ័យបីរយម៉ឺនបួន"
assert tha.decimals.processor("123.001") == "មួយរយវិញ្ញ័យបីរយសុទ្ធមួយ"
assert tha.decimals.processor("-123.0012") == "ដកមួយរយវិញ្ញ័យបីរយសុទ្ធដប់ពីរ"
assert tha.decimals.processor("-123,0012") == "ដកមួយរយវិញ្ញ័យបីរយសុទ្ធដប់ពីរ"

## Number - Ordinals
assert tha.ordinals.processor("5th") == "ទីប្រាំ"
assert tha.ordinals.processor("3rd") == "ទីបី"
assert tha.ordinals.processor("1st") == "ទីមួយ"
assert tha.ordinals.processor("10th") == "ទីដប់"
assert tha.ordinals.processor("10") == "10"

## Number - Currency
assert tha.currency.processor("$100.01") == "មួយរយដុល្លារមួយសេន"
assert tha.currency.processor("$100") == "មួយរយដុល្លារ"
assert tha.currency.processor("100$") == "មួយរយដុល្លារ"
assert tha.currency.processor("100៛") == "មួយរយរៀល"
assert tha.currency.processor("100.32៛") == "មួយរយបីរយសាមសិបពីររៀល"
assert tha.currency.processor("100.0032៛") == "មួយរយបីរយសាមសិបពីររៀល"
```

Homograph Disambiguation

លោក តារា បានប្រកួតឈ្នះ លោក សុខចំនួន 4-1

តើ 4-1 ស្មើប៉ុន្មាន?

ទឹកកាន់តែឡើងបន្តិចម្តងៗ

iPhone 8, iPhone 11, iPhone 12

ស.ជ.ណ, គជប, ខ្ញុំឈឺក, លើកដៃ

Khmer Phoneme Inventory

	Bilabial	Alveolar	Palatal	Velar	Glottal
Plosives	/p/	/t/	/c/	/k/	/ʔ/
Asp. Plosives	/p ^h -/	/t ^h -/	/c ^h -/	/k ^h -/	
Implosives	/b-/	/d-/			
Fricatives		/s-/			/h/
Nasals	/m/	/n/	/ɲ/	/ŋ/	
Semi-vowels	/w/		/j/		
Lateral		/l/			
Flap		/r-/			

[Khmer Character Specification/Usages – Makara Sok et. al.](#)

Pronunciation Dictionary

111,000+ Pairs
Khmer + English

បង់ដា	baŋ.daa
បង់តូស	baŋ.tooh
បង់ទុយ	baŋ.tuj
បង់បក់	baŋ.bak
បង់បត់	baŋ.bat
បង់បាញ់	baŋ.ban
បង់បោយ	baŋ.baoj
បង់ពន្ធ	baŋ.pun
បង់មៀត	baŋ.msiet
បង់សូអាត	baŋ.soo.?aat
បង់សូអ៊ុច	baŋ.soo.?ic
បង់សៀត	baŋ.siet
បង់សែន	baŋ.saen
បង់ស៊ីនប៉េនីស៊ីលីន	baŋ.siin.pee.nii.sii.liin
បង់ស៊ីស	baŋ.siih

Phoneme Generator

Lexicon 69k records

ចាក្រំ → c a k

ចក្រំ → c a k

អ្នក → n e a k

នាក់ → n e a k

...

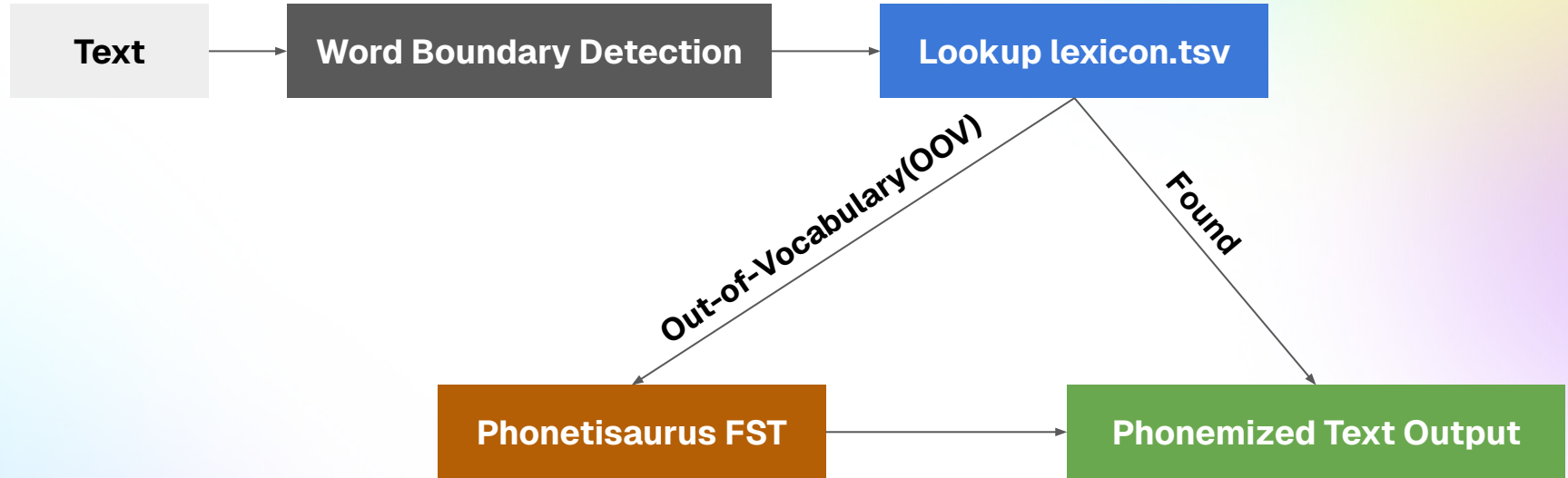
Train

Phonetisaurus

Finite State
Transducer Model

model.fst

Text Phonemicization



Model Architecture Comparison

Architecture	Type	Speed	Key Features	Strengths	Weaknesses
Tacotron 2	Autoregressive seq2seq	Slow	Encoder-decoder with attention, generates mel-spectrograms	High naturalness, expressive speech, well-documented	Slow inference, can skip/repeat words, attention alignment issues
FastSpeech / FastSpeech 2	Non-autoregressive feed-forward	Very Fast (270x faster than Tacotron)	Parallel generation, duration predictor, transformer-based	Extremely fast, robust (no word skipping), controllable speed	Requires teacher model or forced alignment for training
VITS	End-to-end VAE-based	Fast	Variational autoencoder, monotonic alignment search, single-stage	Direct text-to-waveform, no separate vocoder needed, efficient	Complex architecture, GPU memory intensive
Transformer TTS	Autoregressive transformer	Medium	Multi-head attention, transformer encoder-decoder	Better than RNN-based models, parallelizable training	Still slower than non-autoregressive models

Dataset Requirements

- ❑ Clean Speech, No background noise, Consistent recording environment
- ❑ Sample Rate: **22.05 kHz → 44.1 kHz**
- ❑ Bit Depth: 16 bits per sample
- ❑ Channel: Mono
- ❑ Audio Format: WAV, FLAC (Lossless)
- ❑ Audio Max Duration: 10 seconds

Audio



Text អ្នកស្រី កែវ ចន្ទ បេទុបនលក់អស់៣០០តោនរួចហើយ ជាមួយតម្លៃ១៥០០០រៀលក្នុងមួយគីឡូក្រាម

Hardware Requirements

CPU: Good enough

GPU VRAM: 24GB Min.

RAM: 32GB

Disk: 1TB

Duration: A week or two (depends on dataset size)

Knowledge Requirements

- ❑ Python / Virtualenv / Anaconda
- ❑ Machine Learning Concepts
- ❑ PyTorch
- ❑ Scipy
- ❑ Numpy
- ❑ C/C++ for binding
- ❑ etc.

Evaluation Metrics

MOS (Mean Opinion Score)

Mean Opinion Score (MOS) for TTS is a subjective evaluation method where human listeners rate synthesized speech on a scale (typically 1-5) for quality, naturalness, and clarity.

CMOS (Comparative Mean Opinion Score)

CMOS, or Comparative Mean Opinion Score, is a method for evaluating TTS quality by having human listeners directly compare two speech samples and choose which one is better.

Perceptual evaluation of speech quality (PESQ)

An objective, algorithmic method for assessing speech quality by comparing a degraded signal to a reference signal, making it applicable to Text-to-Speech (TTS) systems.

Deployment

TorchServe

ONNXRuntime by Microsoft (Anywhere)

NCNN from Tencent (Optimized for Mobile)

MNN from Alibaba (Optimized for Mobile)

Triton Inference Server by NVIDIA (NVIDIA Chips)

Demo

<https://vok-tts-samples.netlify.app/>

<https://sarika.gov.kh>

Next Challenges

- Khmer Word Tokenizer
- Neural Verbalizer
- Better Punctuation Model

References

<https://github.com/seanghay/awesome-khmer-language>

<https://github.com/AdolfVonKleist/Phonetisaurus>

<https://github.com/seanghay/khmernormalizer>

<https://github.com/seanghay/khmercut>

<https://github.com/seanghay/tha>

<https://github.com/seanghay/phonetisaurus-is>

<https://github.com/sillsdev/khmer-character-specification>

<https://github.com/seanghay/automatic-phonemic-and-phonetic-transcription>

Questions

Email: seanghay.dev@gmail.com

GitHub: @seanghay

Twitter: @seanghay_yath

Telegram:

https://t.me/seanghay_yath