What is
"end-to-end" text-to-speech synthesis ?

Simon King, Centre for Speech Technology Research, University of Edinburgh, UK

Deep Voice: Real-time Neural Text-to-Speech

Sercan Ö. Arık [*1] **Mike Chrzanowski** [*1] **Adam Coates** [*1] **Gregory Diamos** [*1] **Andrew Gibiansky** [*1]
Yongguo Kang [*2] **Xian Li** [*2] **John Miller** [*1] **Andrew Ng** [*1] **Jonathan Raiman** [*1] **Shubho Sengupta** [*1]
Mohammad Shoeybi [*1]

Abstract

We present Deep Voice, a production-quality text-to-speech system constructed entirely from deep neural networks. Deep Voice lays the groundwork for truly end-to-end neural speech synthesis. The system comprises five major building blocks: a segmentation model for locating phoneme boundaries, a grapheme-to-phoneme conversion model, a phoneme duration prediction model, a fundamental frequency prediction model, and an audio synthesis model.

Fundamentally, it allows human-technology interaction without requiring visual interfaces. Modern TTS systems are based on complex, multi-stage processing pipelines, each of which may rely on hand-engineered features and heuristics. Due to this complexity, developing new TTS systems can be very labor intensive and difficult.

Deep Voice is inspired by traditional text-to-speech pipelines and adopts the same structure, while replacing all components with neural networks and using simpler features: first we convert text to phoneme and then use an audio synthesis model to convert linguistic features into

INTERSPEECH 2017
August 20–24, 2017, Stockholm, Sweden

Tacotron: Towards End-to-End Speech Synthesis

Yuxuan Wang, RJ Skerry-Ryan*, Daisy Stanton, Yonghui Wu, Ron J. Weiss†, Navdeep Jaitly, Zongheng Yang, Ying Xiao*, Zhifeng Chen, Samy Bengio†, Quoc Le, Yannis Agiomyrgiannakis, Rob Clark, Rif A. Saurous**

Google, Inc.

{yxwang, rjryan, rif}@google.com

Abstract

A text-to-speech synthesis system typically consists of multiple stages, such as a text analysis frontend, an acoustic model and an audio synthesis module. Building these components often requires extensive domain expertise and may contain brittle

this is a particularly difficult learning task for an end-to-end model: it must cope with large variations at the signal level for a given input. Moreover, unlike end-to-end speech recognition [4] or machine translation [5], TTS outputs are continuous, and output sequences are usually much longer than those of the input. These attributes cause prediction errors to accu-

Published as a conference paper at ICLR 2018

DEEP VOICE 3: SCALING TEXT-TO-SPEECH WITH CONVOLUTIONAL SEQUENCE LEARNING

Wei Ping*, **Kainan Peng***, **Andrew Gibiansky***, **Sercan Ö. Arık***
Ajay Kannan, **Sharan Narang**
Baidu Research
{pingwei01, pengkainan, gibianskyandrew, sercanarik, kannanajay, sharan}@baidu.com

Jonathan Raiman*†
OpenAI
raiman@openai.com

John Miller*†
University of California, Berkeley
miller_john@berkeley.edu

DOI: 10.1109/ICASSP.2018.8461368 — Tacotron 2

NATURAL TTS SYNTHESIS BY CONDITIONING WAVENET ON MEL SPECTROGRAM PREDICTIONS

Jonathan Shen[1], *Ruoming Pang*[1], *Ron J. Weiss*[1], *Mike Schuster*[1], *Navdeep Jaitly*[1], *Zongheng Yang*[*2], *Zhifeng Chen*[1], *Yu Zhang*[1], *Yuxuan Wang*[1], *RJ Skerry-Ryan*[1], *Rif A. Saurous*[1], *Yannis Agiomyrgiannakis*[1], *and Yonghui Wu*[1]

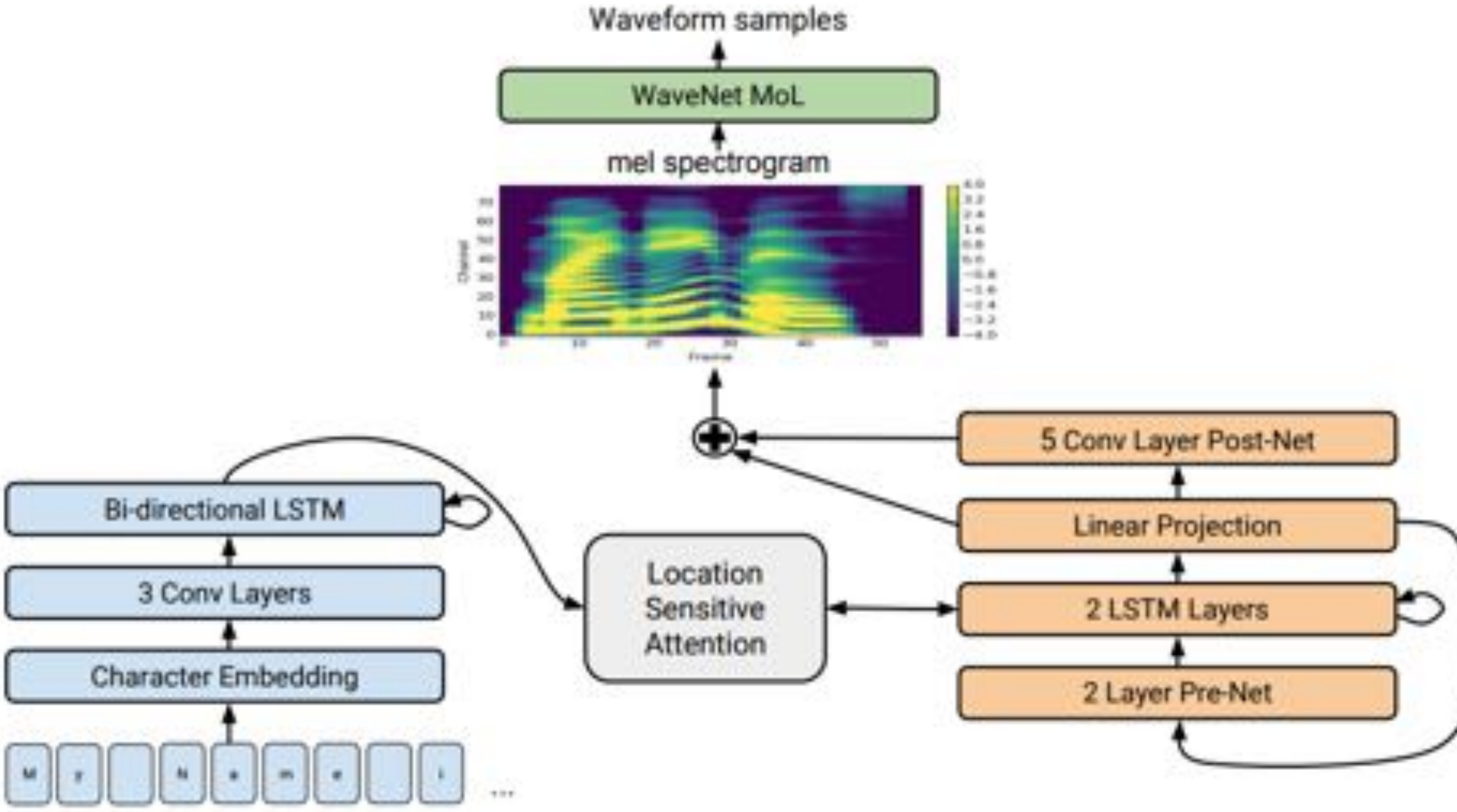
[1]Google, Inc., [2]University of California, Berkeley,
{jonathanasdf,rpang,yonghui}@google.com

ABSTRACT

This paper describes Tacotron 2, a neural network architecture for speech synthesis directly from text. The system is composed of a recurrent sequence-to-sequence feature prediction network that maps character embeddings to mel-scale spectrograms, followed by a modified WaveNet model acting as a vocoder to synthesize time-domain waveforms from those spectrograms. Our model achieves a mean opinion score (MOS) of 4.53 comparable to a MOS of 4.58 for professionally recorded speech. To validate our design choices, we present

the authors note, this was simply a placeholder for future neural vocoder approaches, as Griffin-Lim produces characteristic artifacts and lower audio quality than approaches like WaveNet.

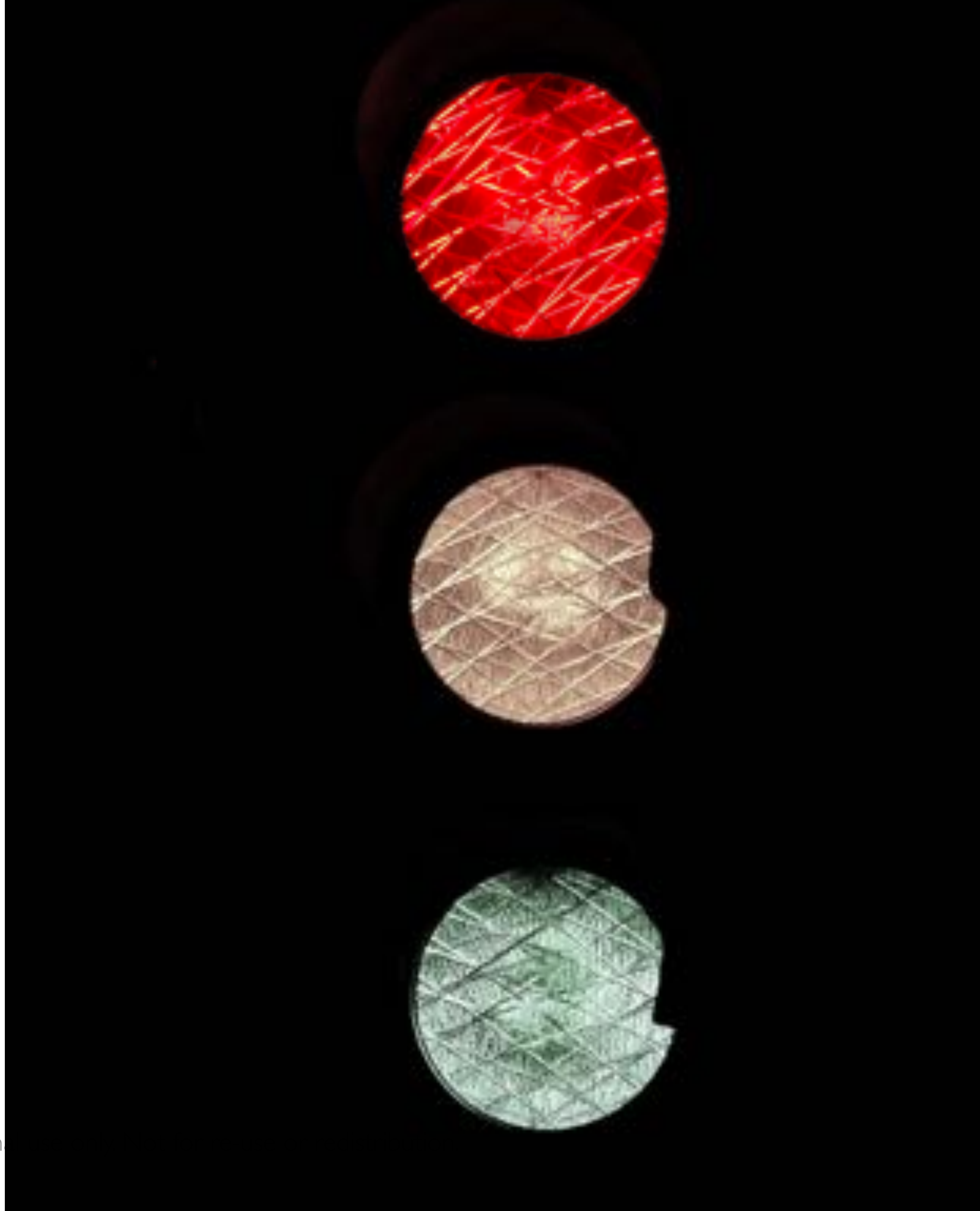
In this paper, we describe a unified, entirely neural approach to speech synthesis that combines the best of the previous approaches: a sequence-to-sequence Tacotron-style model [12] that generates mel spectrograms, followed by a modified WaveNet vocoder [10, 15]. Trained directly on normalized character sequences and corresponding speech waveforms, our model learns to synthesize natural sounding speech that is difficult to distinguish from real human speech.



Waveform samples
WaveNet MoL
mel spectrogram
5 Conv Layer Post-Net
Linear Projection
2 LSTM Layers
2 Layer Pre-Net
Location Sensitive Attention
Bi-directional LSTM
3 Conv Layers
Character Embedding

Outline

- Tutorial
  - Text processing
  - Regression
  - Waveform generation
- Current research
  - Waveform generation
  - Regression
  - Text processing
- What next?



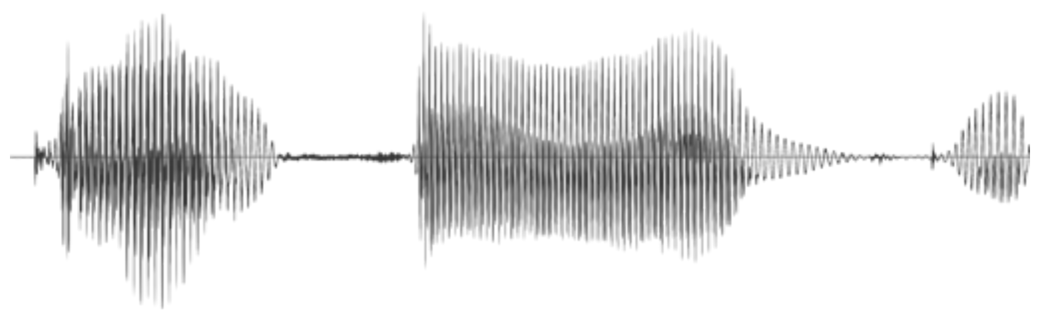
The end-to-end problem we want to solve

Text-to-Speech

text

waveform

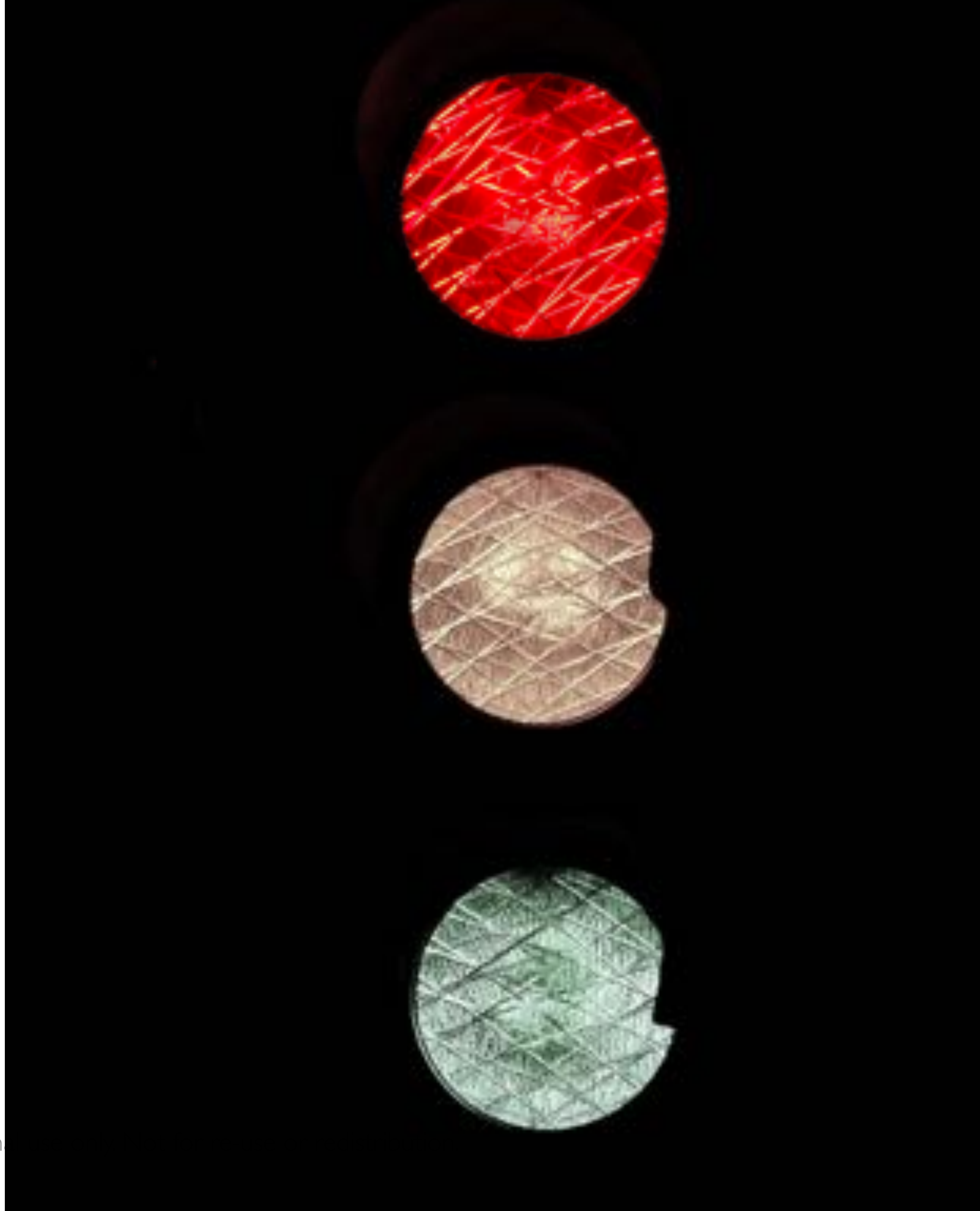
`Author of the…`



The three-stage pipeline of text-to-speech synthesis (TTS)

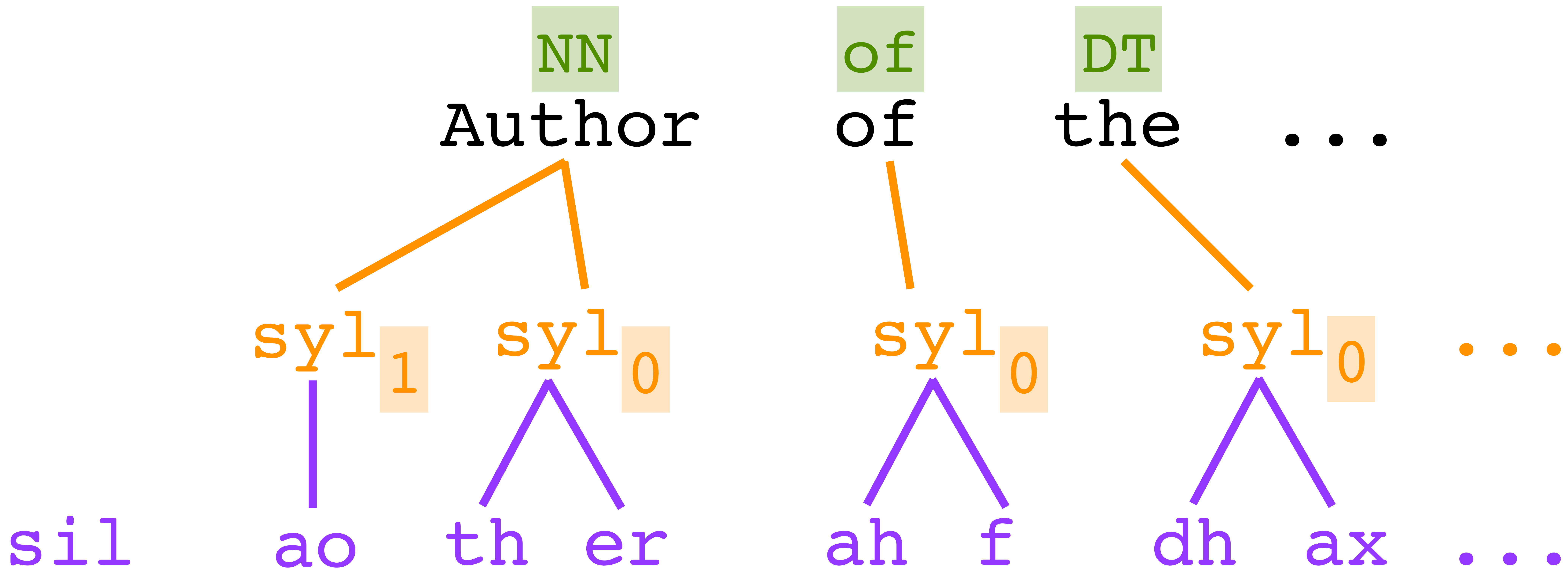
Outline

- Tutorial
  - Text processing
  - Regression
  - Waveform generation
- Current research
  - Waveform generation
  - Regression
  - Text processing
- What next?

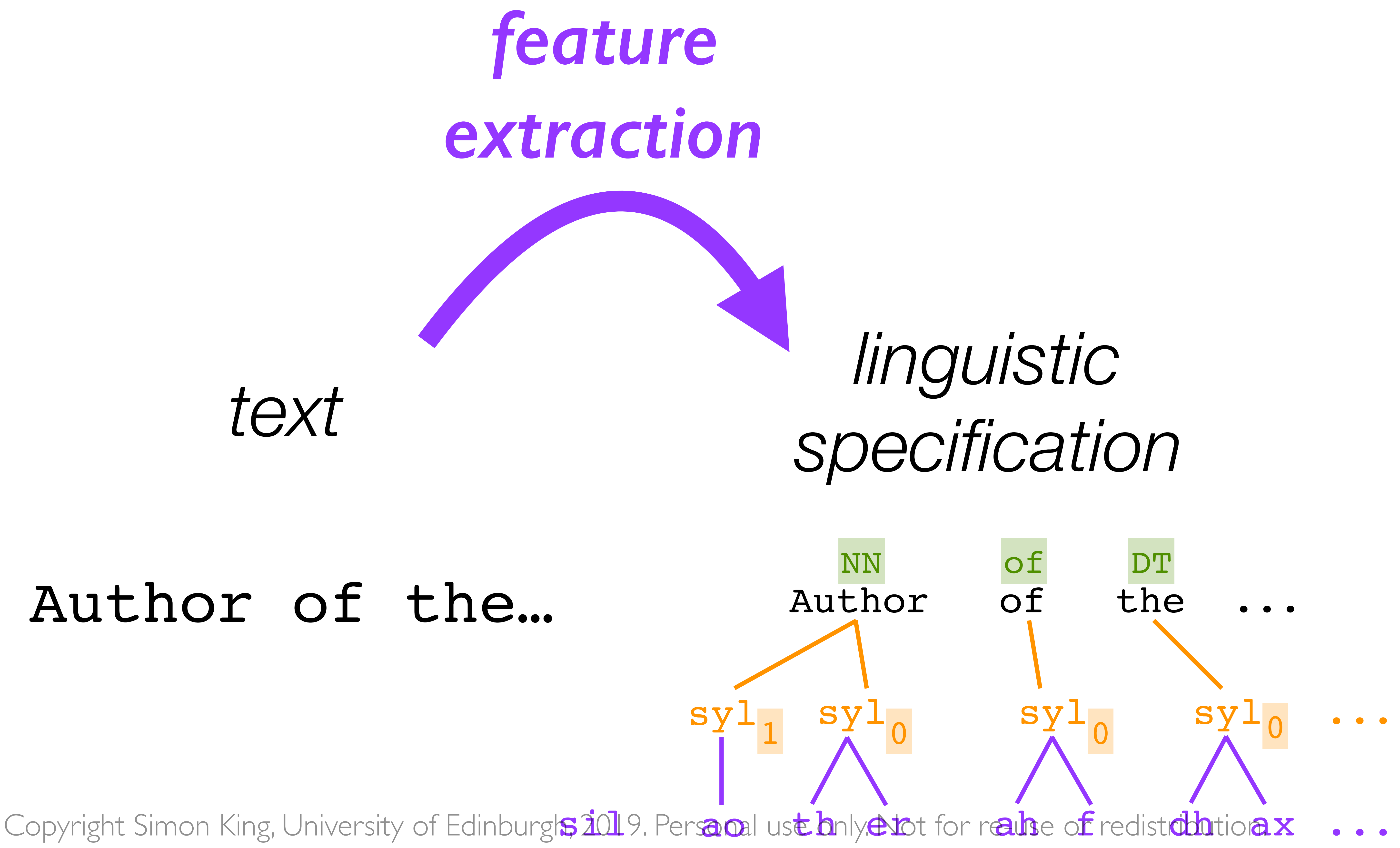


Text processing in the "front end"

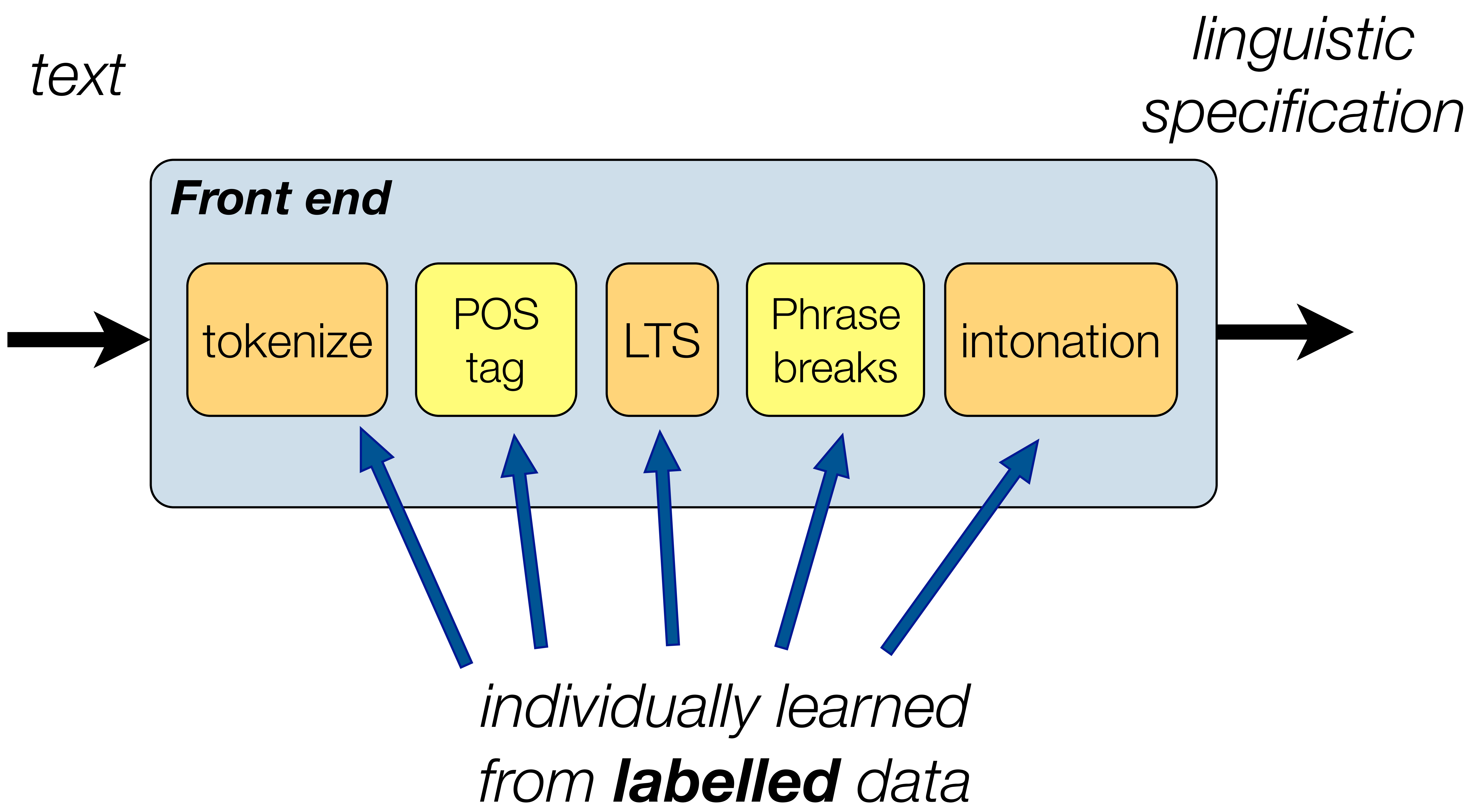
The linguistic specification



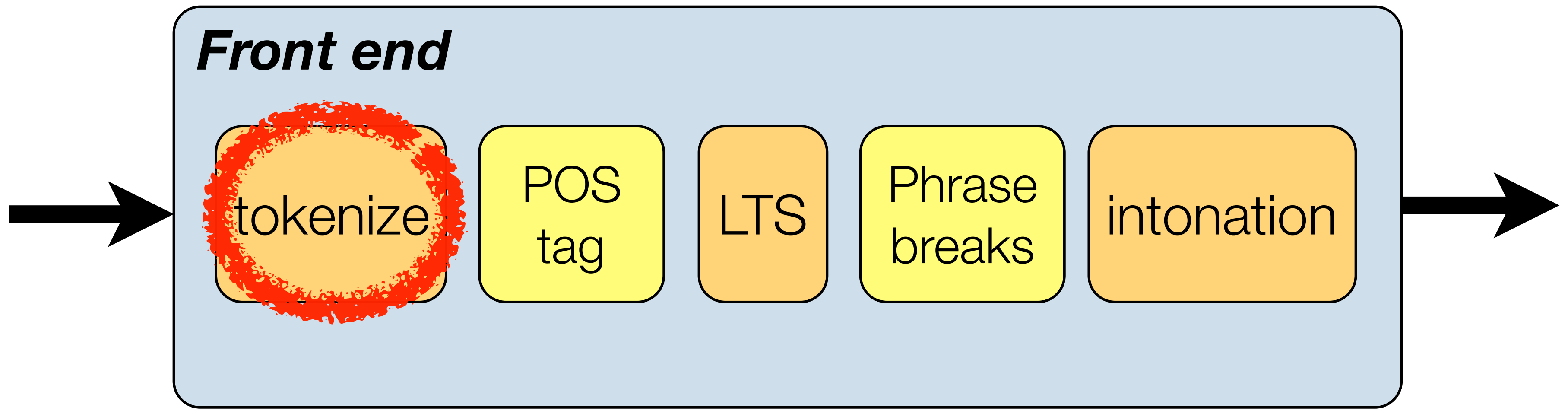
Extracting features from text using the front end



Text processing pipeline



Tokenize & Normalize

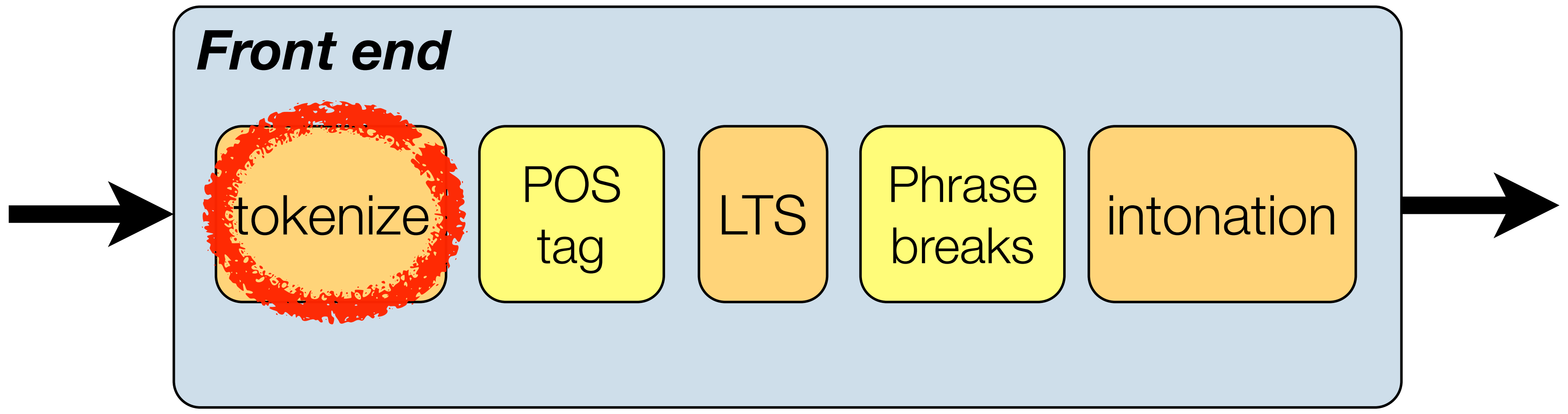


- Step 1: divide input stream into tokens, which are potential words

- For English and many other languages
  - rule based
  - whitespace and punctuation are good features

- For some other languages, especially those that don't use whitespace
  - may be more difficult
  - other techniques required (out of scope here)

Tokenize & Normalize

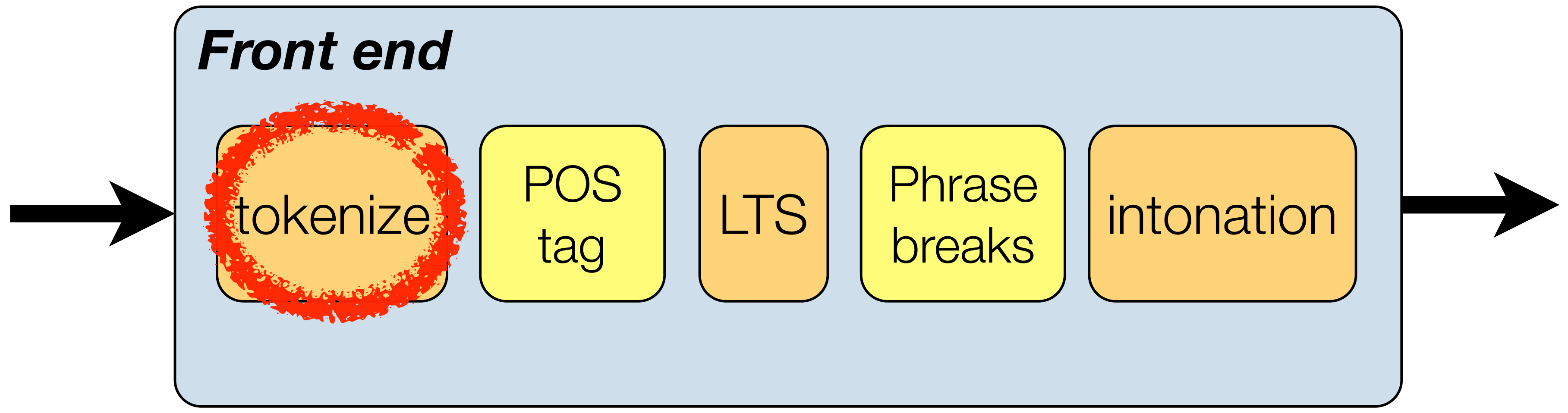


- Step 2: classify every token, finding **Non-Standard Words** that need further processing

`In 2011, I spent £100 at IKEA on 100 DVD holders.`

`NYER MONEY ASWD NUM LSEQ`

Tokenize & Normalize



- Step 3: a set of specialised modules to process NSWs of a each type

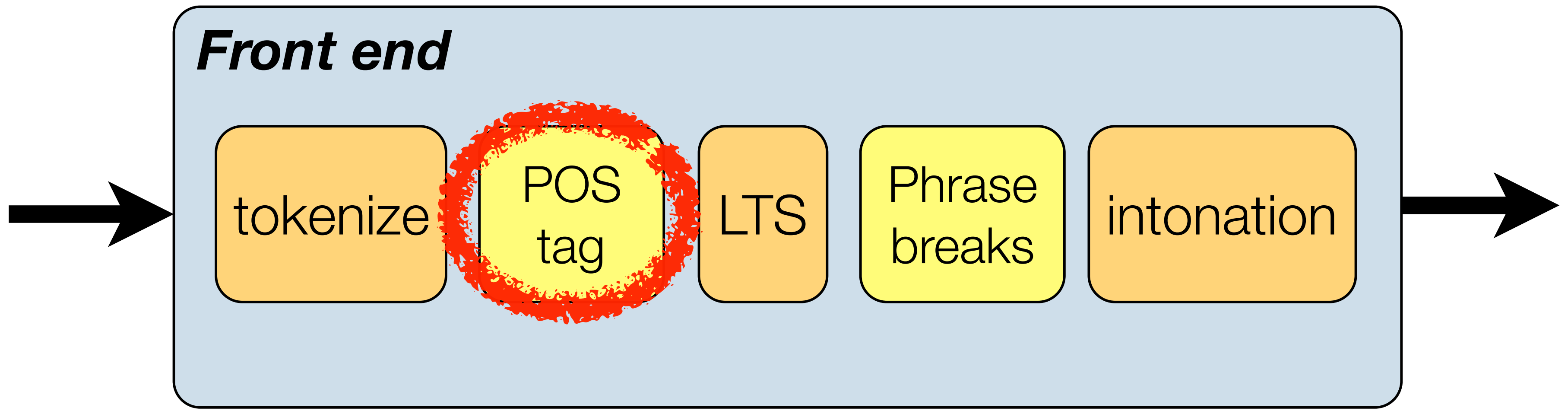
2011 ⇨ NYER ⇨ twenty eleven

£100 ⇨ MONEY ⇨ one hundred pounds

IKEA ⇨ ASWD ⇨ *apply letter-to-sound*

100 ⇨ NUM ⇨ one hundred

DVD ⇨ LSEQ ⇨ D. V. D. ⇨ dee vee dee

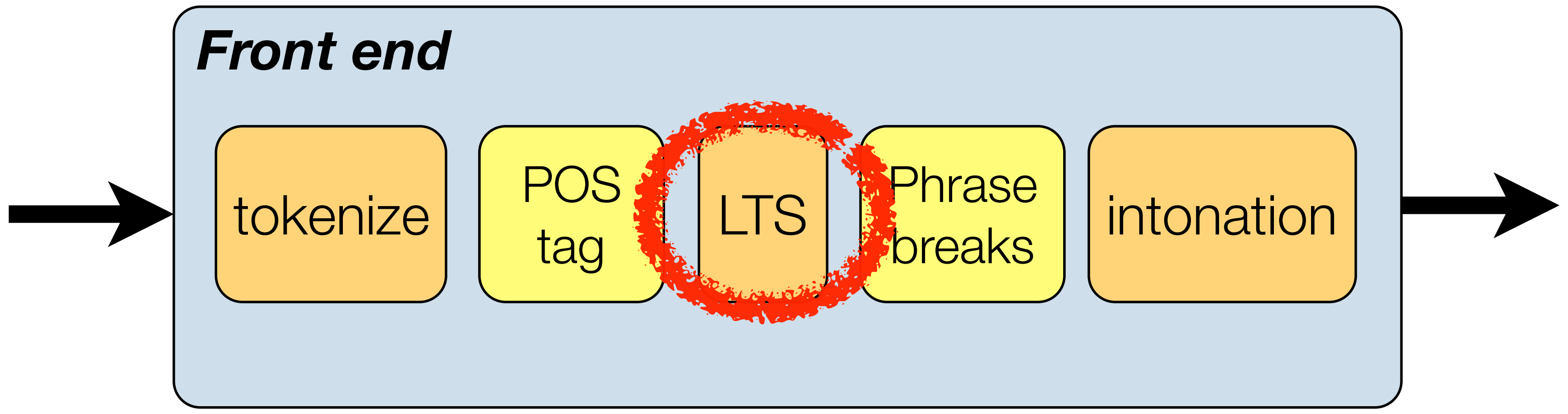


POS tagging

- Part-of-speech tagger
- Accuracy can be very high
- Trained on **annotated** text data
- **Categories** are designed for text, not speech

```
NP  Ed
NP  Beard,
VBZ says
DT  the
NN  push
IN  for
VBP do
PP  it
PP  yourself
NN  lawmaking
VBZ comes
IN  from
NNS voters
WP  who
VBP feel
VBN frustrated
IN  by
PP$ their
JJ  elected
NNS officials.
CC  But
DT  the
NN  initiative
```

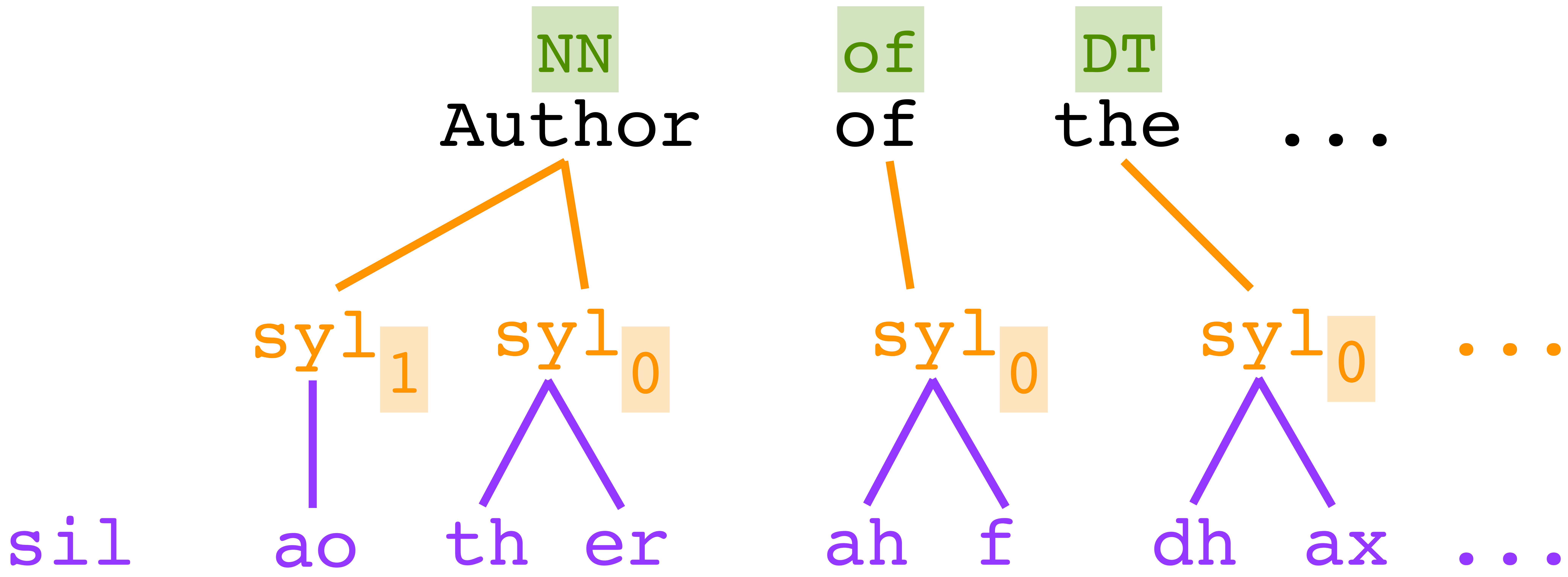
Pronunciation / LTS



- Pronunciation model
  - dictionary look-up, *plus*
  - letter-to-sound model
- But
  - need deep **knowledge** of the language to design the phoneme set
  - human **expert** must write dictionary

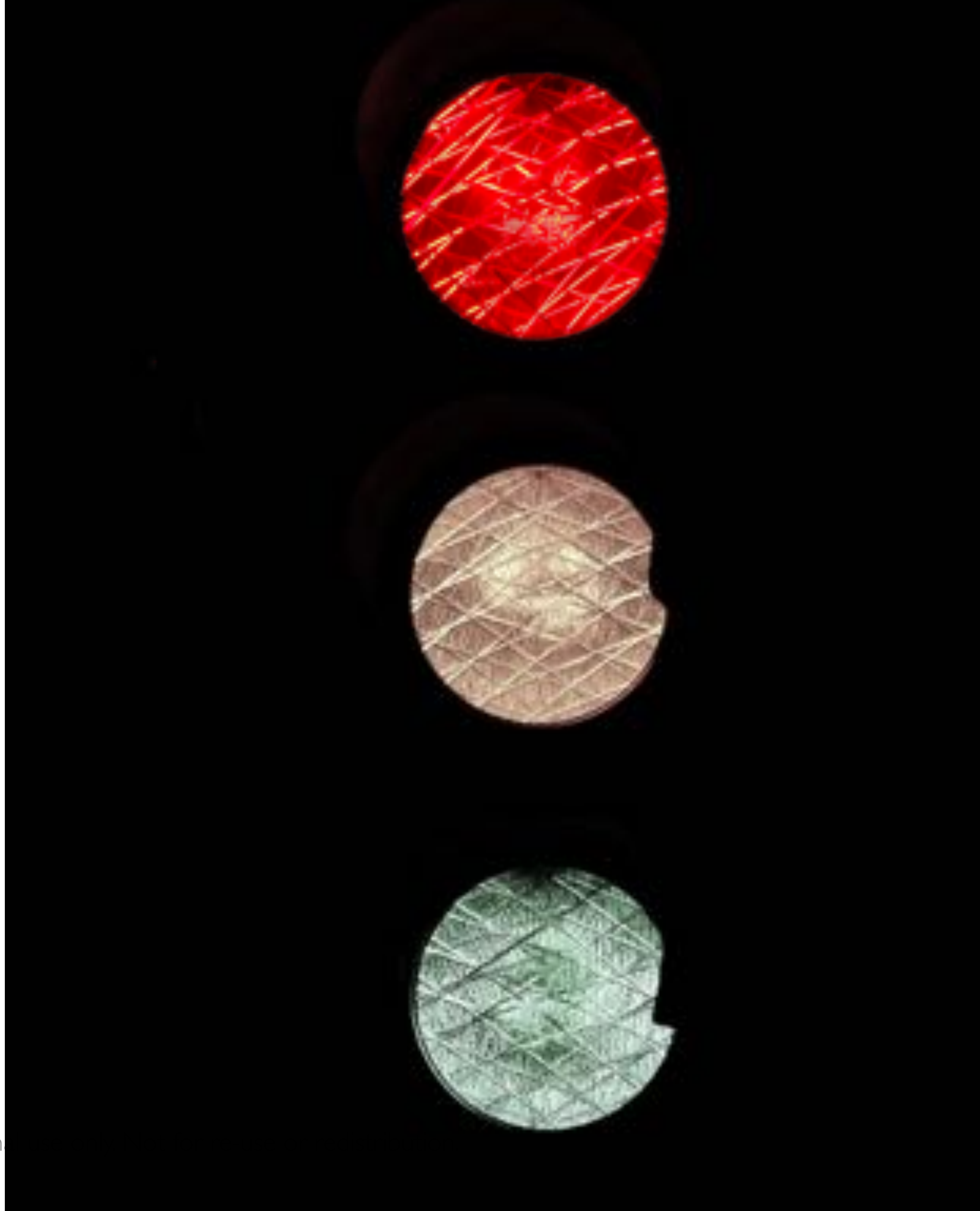
```
AERIALS   EH1 R IY0 AH0 L Z
AERIE   EH1 R IY0
AERIEN   EH1 R IY0 AH0 N
AERIENS   EH1 R IY0 AH0 N Z
AERITALIA   EH2 R IH0 T AE1 L Y AH0
AERO   EH1 R OW0
AEROBATIC   EH2 R AH0 B AE1 T IH0 K
AEROBATICS   EH2 R AH0 B AE1 T IH0 K S
AEROBIC   EH0 R OW1 B IH0 K
AEROBICALLY   EH0 R OW1 B IH0 K L IY0
AEROBICS   ER0 OW1 B IH0 K S
AERODROME   EH1 R AH0 D R OW2 M
AERODROMES   EH1 R AH0 D R OW2 M Z
AERODYNAMIC   EH2 R OW0 D AY0 N AE1 M IH0 K
AERODYNAMICALLY   EH2 R OW0 D AY0 N AE1 M IH0 K L
AERODYNAMICIST   EH2 R OW0 D AY0 N AE1 M IH0 S IH
AERODYNAMICISTS   EH2 R OW0 D AY0 N AE1 M IH0 S I
AERODYNAMICISTS(1)   EH2 R OW0 D AY0 N AE1 M IH0
AERODYNAMICS   EH2 R OW0 D AY0 N AE1 M IH0 K S
AERODYNE   EH1 R AH0 D AY2 N
AERODYNE'S   EH1 R AH0 D AY2 N Z
AEROFLOT   EH1 R OW0 F L AA2 T
```

The linguistic specification



Outline

- Tutorial
  - Text processing
  - Regression
  - Waveform generation
- Current research
  - Waveform generation
  - Regression
  - Text processing
- What next?



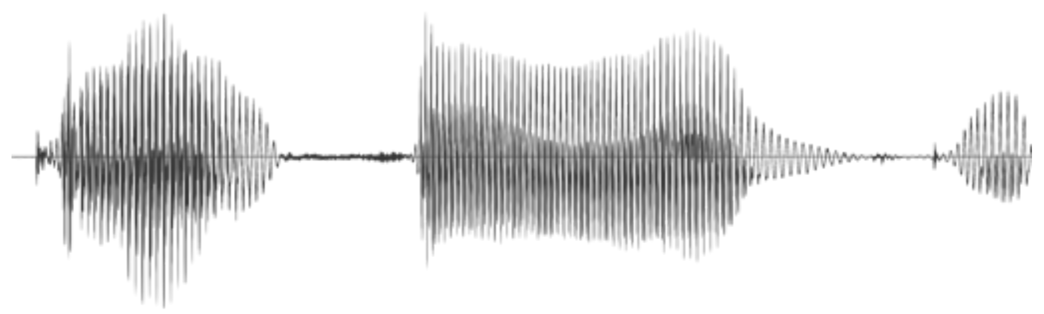
The end-to-end problem we want to solve

Text-to-Speech

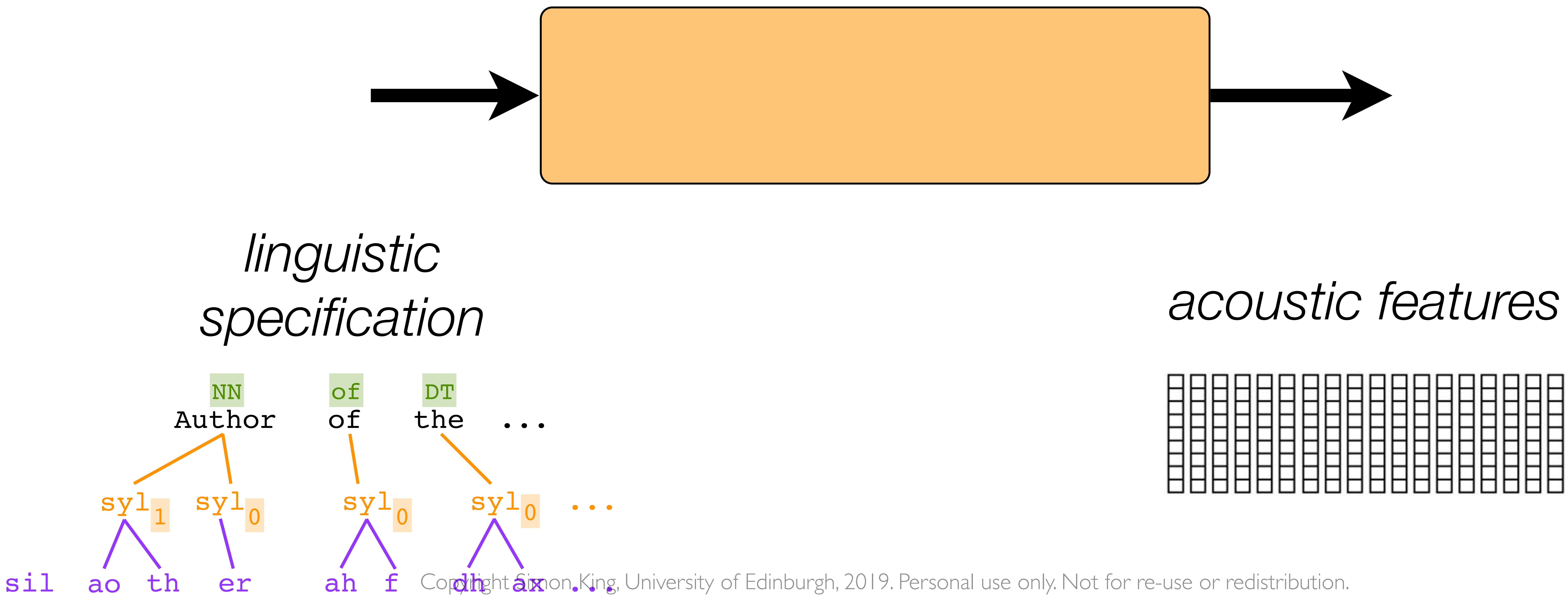
text

waveform

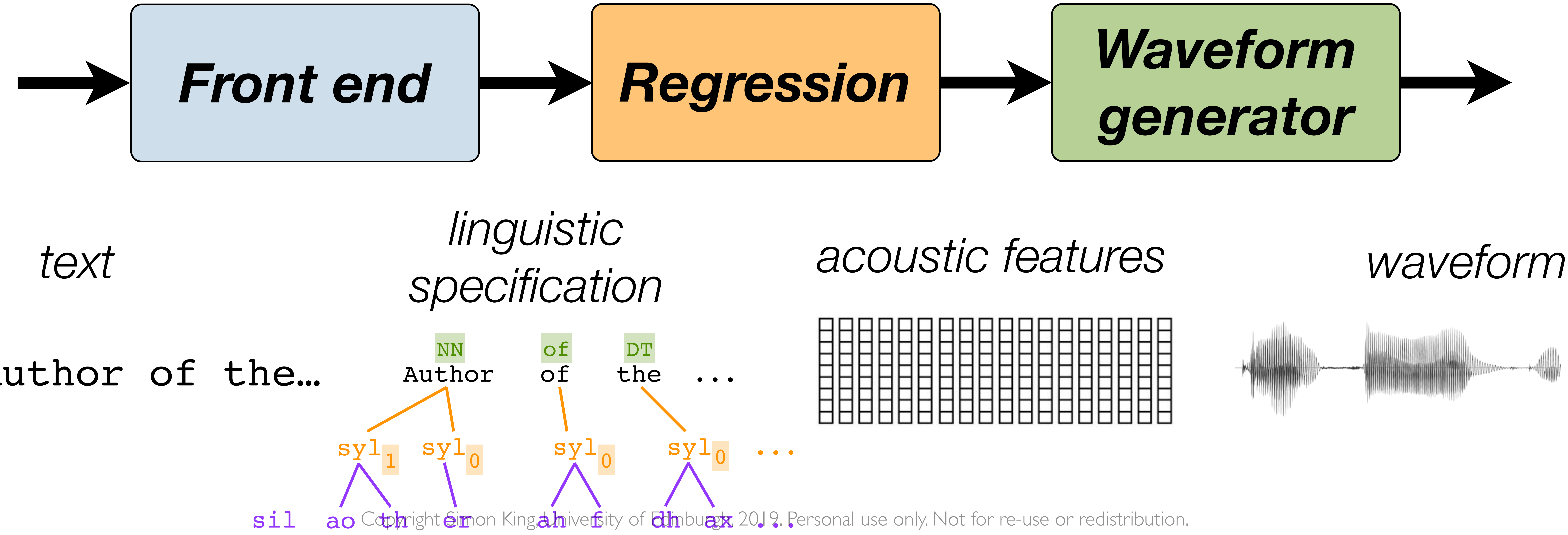
`Author of the…`



A problem we can actually solve (perhaps with machine learning)



The three-stage pipeline of text-to-speech synthesis (TTS)

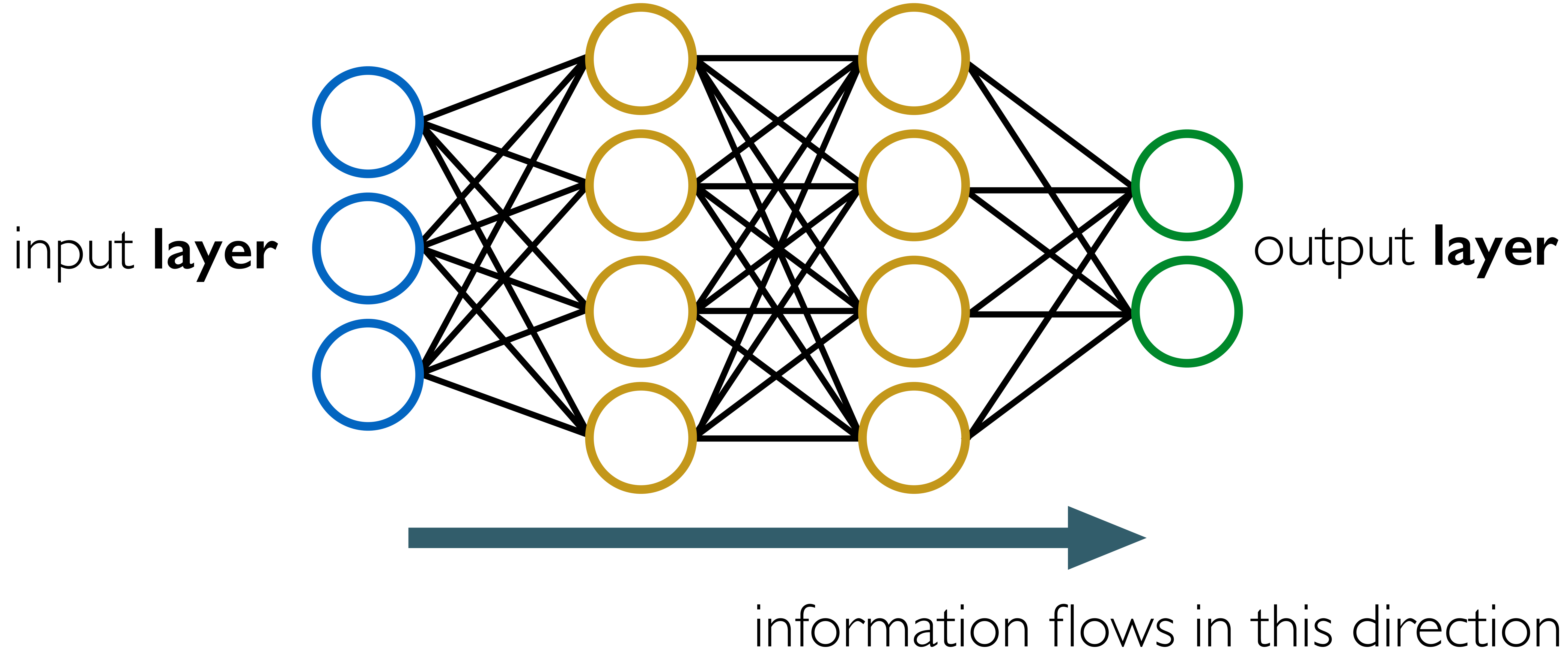


The three-stage pipeline of text-to-speech synthesis (TTS)

Regression predicts acoustic features from linguistic features

A general-purpose regression model: a neural network

Regression

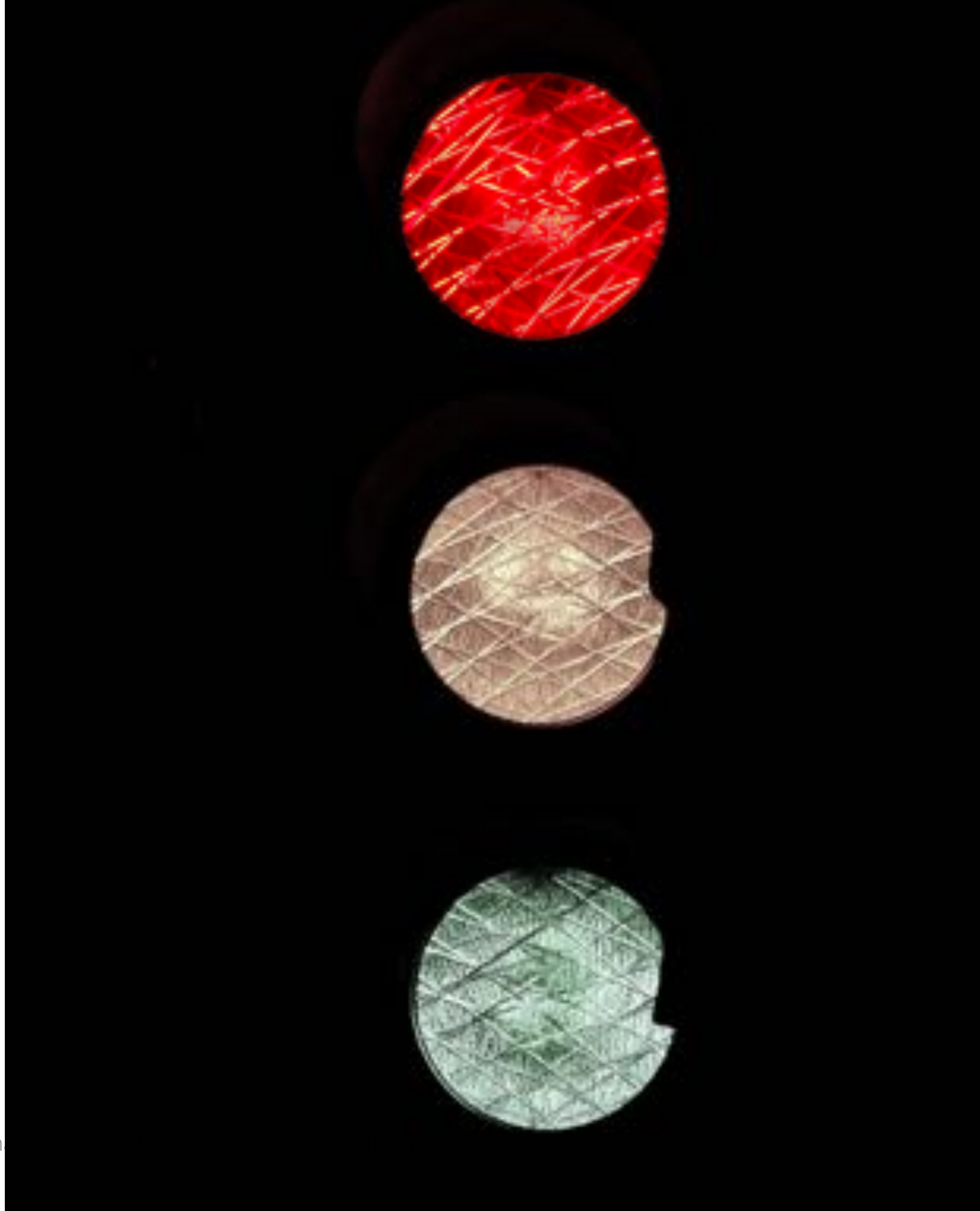


Doing regression with a neural network

Regression

Outline

- Tutorial
  - Text processing
  - Regression
  - <u>Waveform generation</u>
- Current research
  - Waveform generation
  - Regression
  - Text processing
- What next?



Generating the waveform

Waveform generator

statistical parametric
speech synthesis

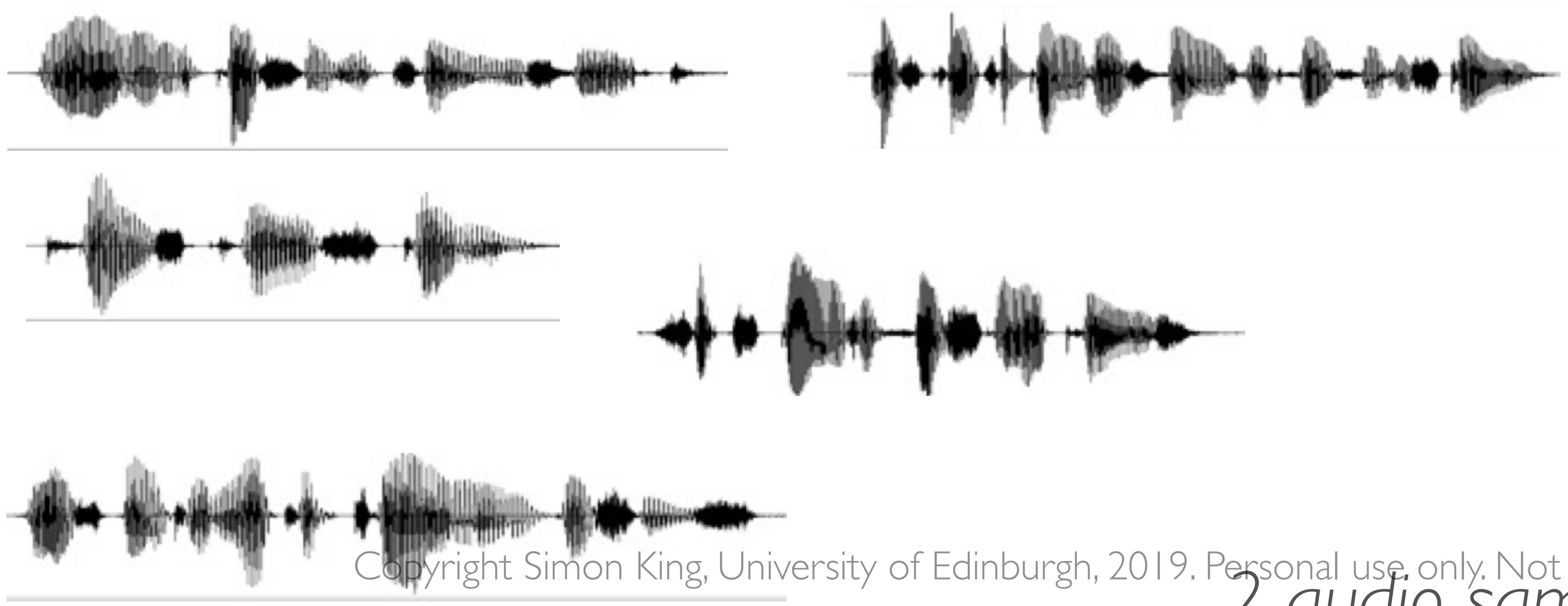
neural speech
synthesis

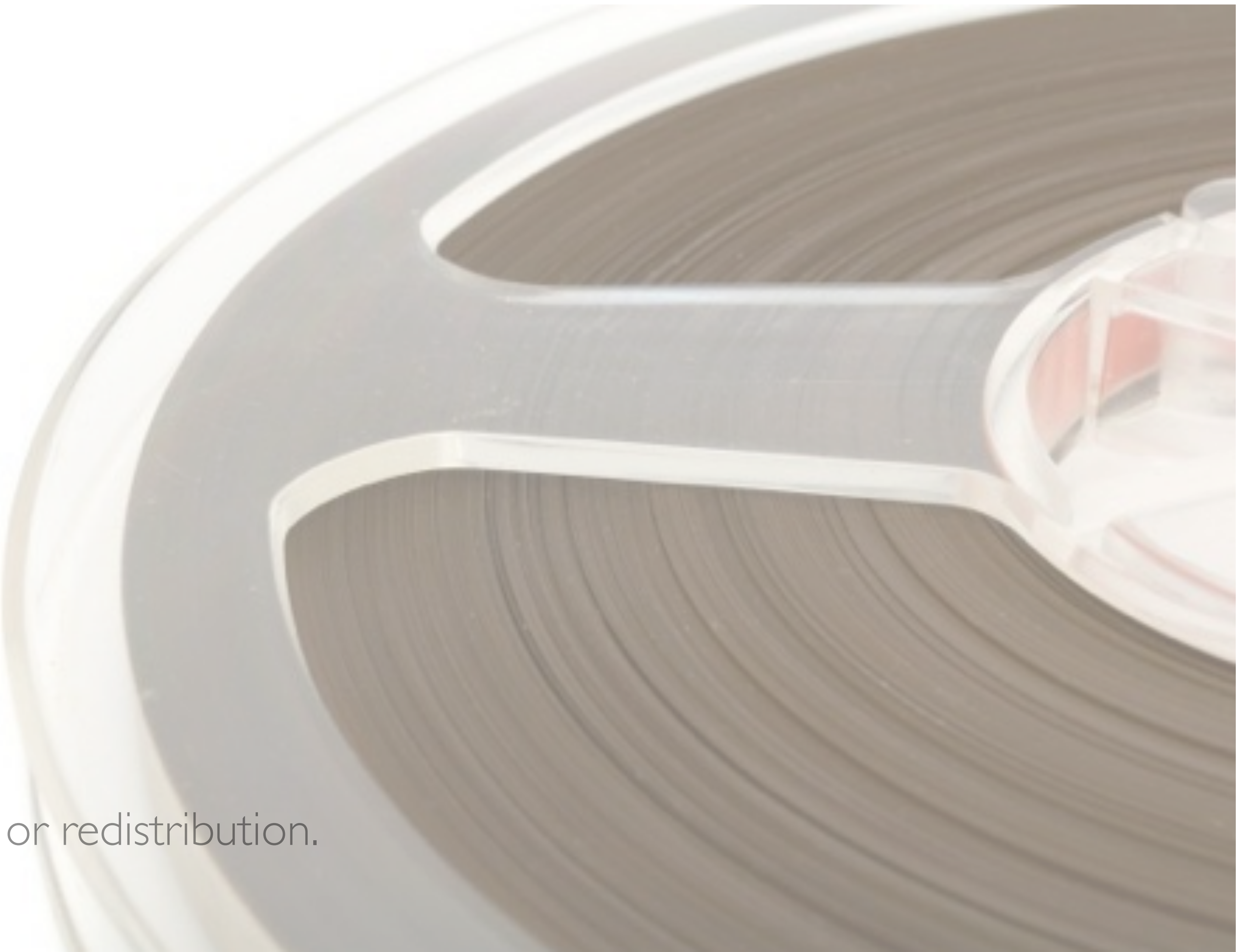
1st generation
unit selection

2nd generation
unit selection

1990 2000 2010 2020

Waveform generator

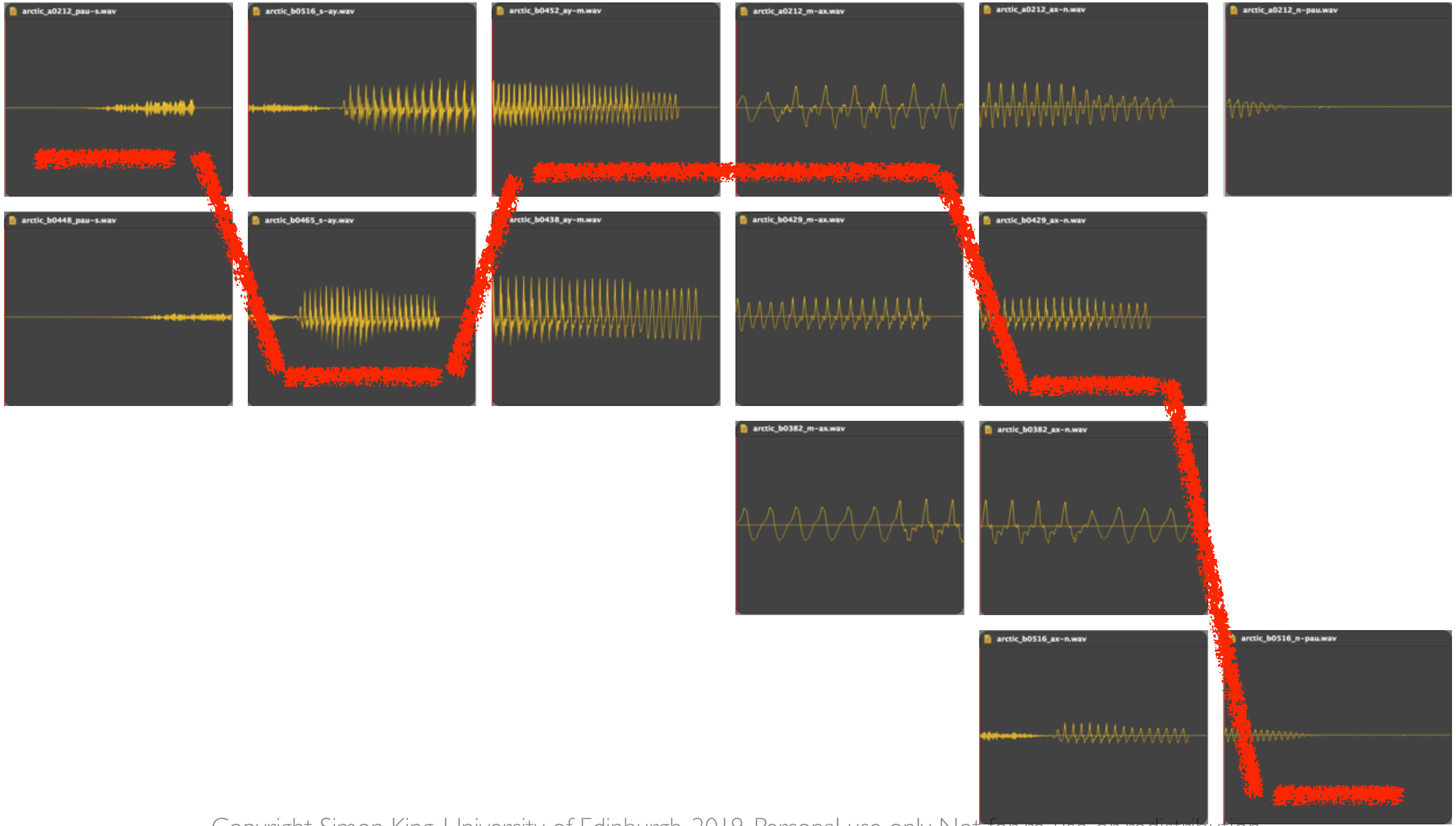




2 audio samples

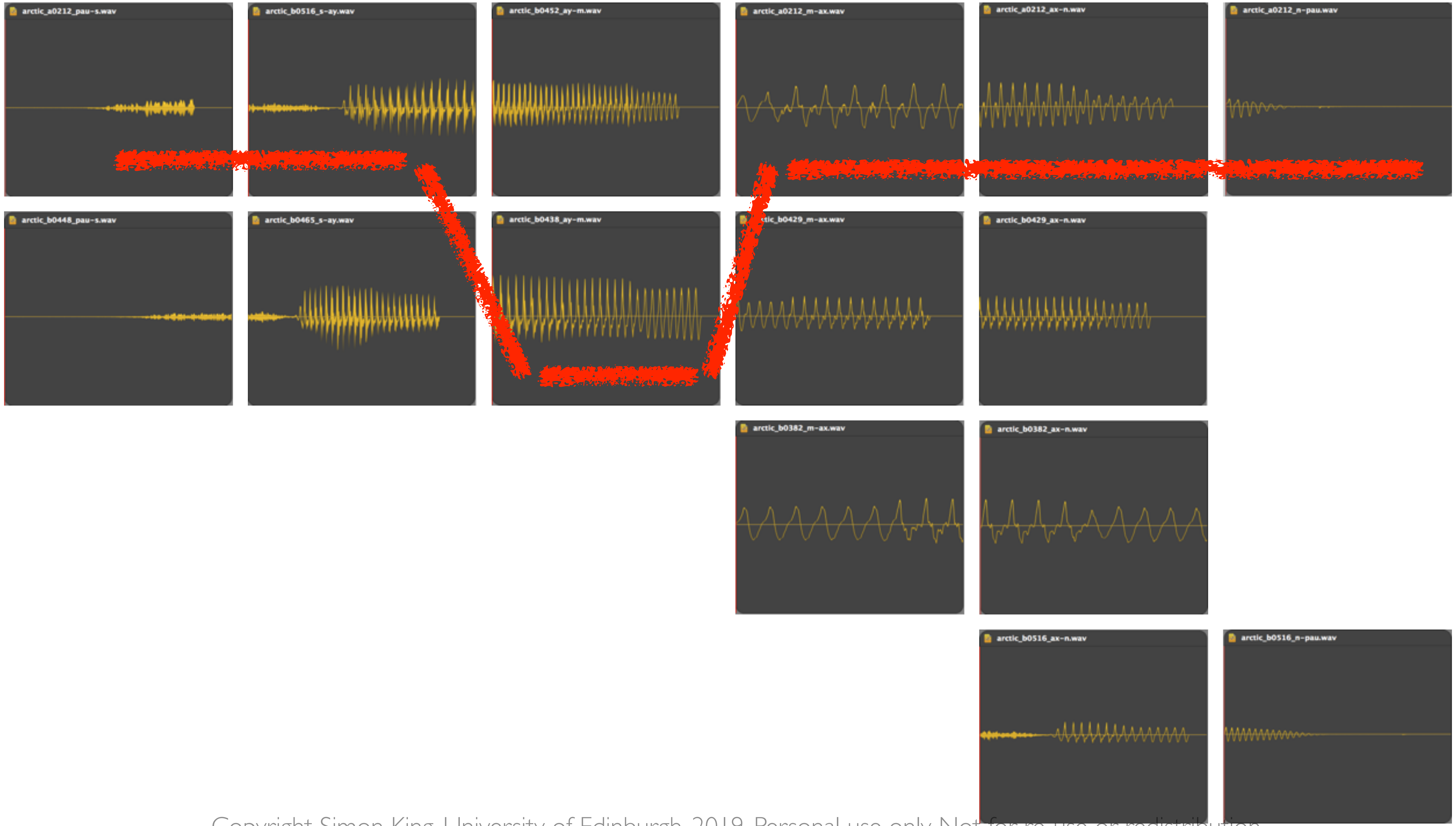
Let's say "Simon"

Waveform generator



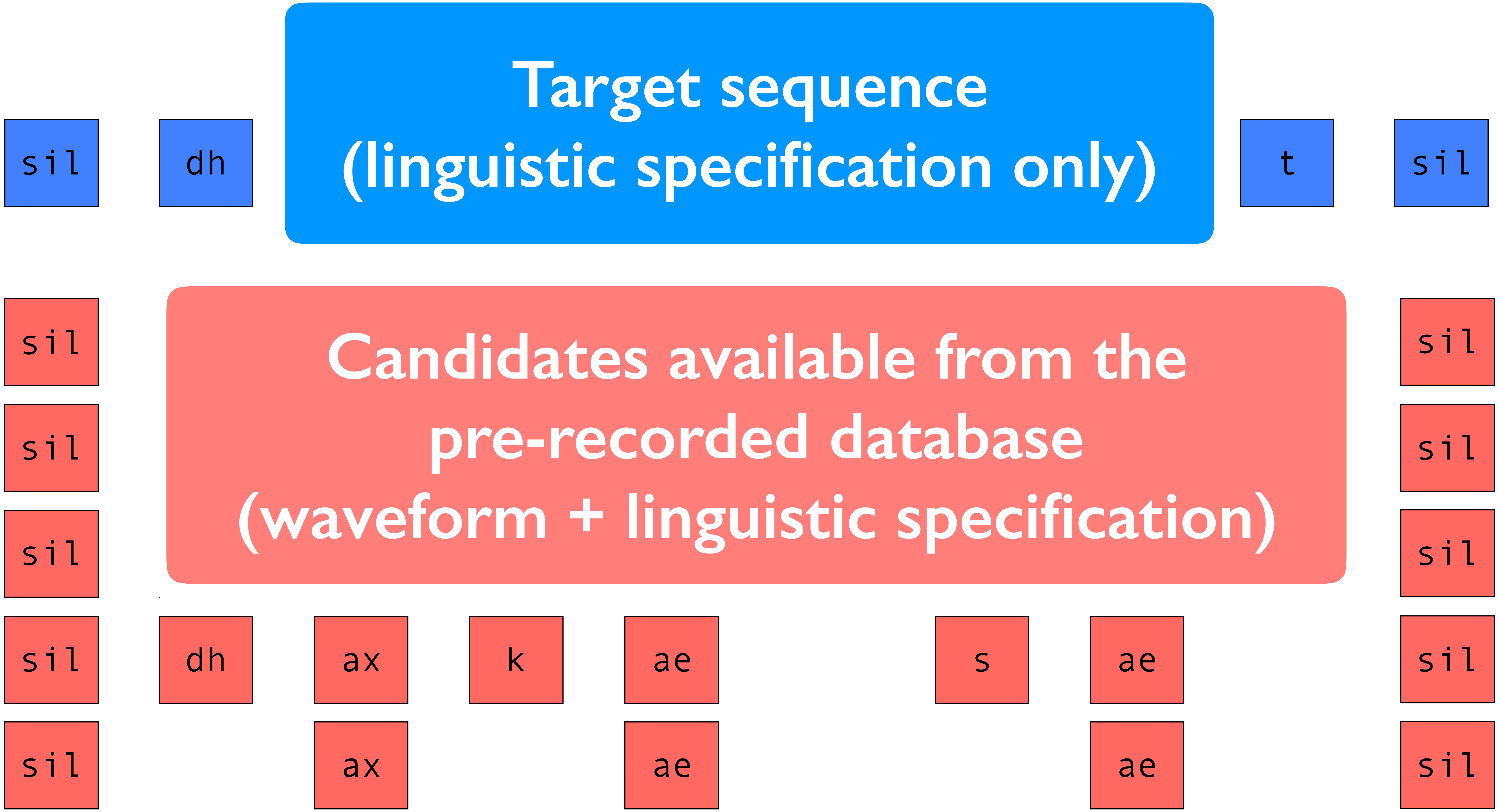
Let's say "Simon"

Waveform generator



Unit selection

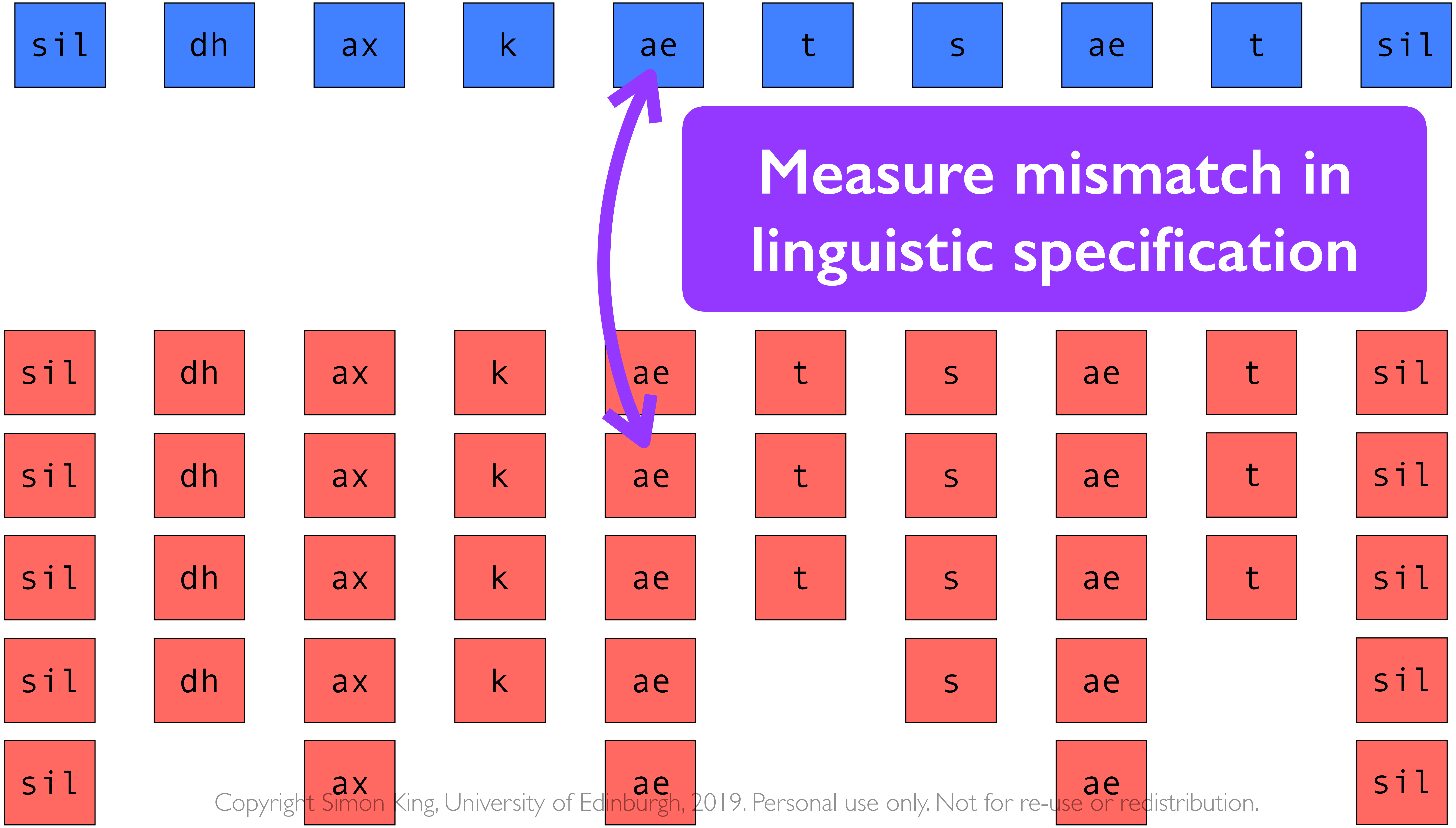
Waveform generator



Unit selection - first generation: using linguistic features directly

Unit selection using only linguistic features

Waveform generator



Waveform generator

statistical parametric
speech synthesis

1st generation
unit selection

2nd generation
unit selection

neural speech
synthesis

1990 2000 2010 2020

Traditional vocoder using signal processing techniques

Waveform generator

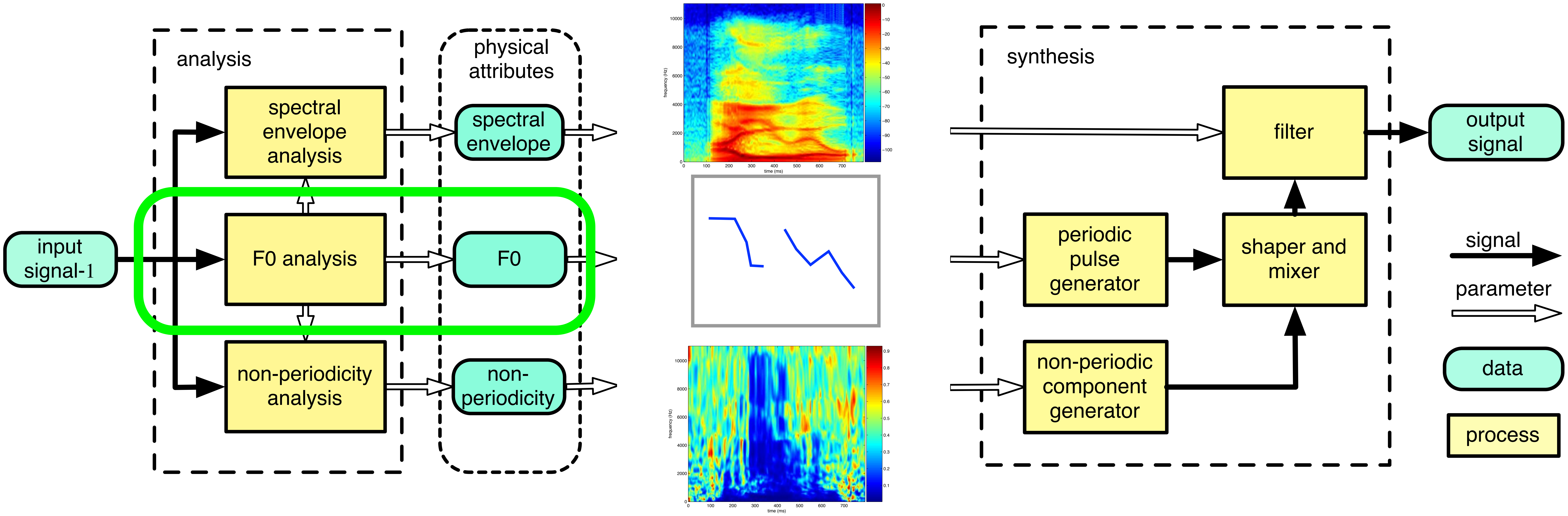


Figure: Hideki Kawahara

Waveform generator

statistical parametric speech synthesis

neural speech synthesis

1st generation unit selection

2nd generation unit selection

1990 2000 2010 2020

Unit selection - second generation: predicting acoustic features

Unit selection using acoustic features

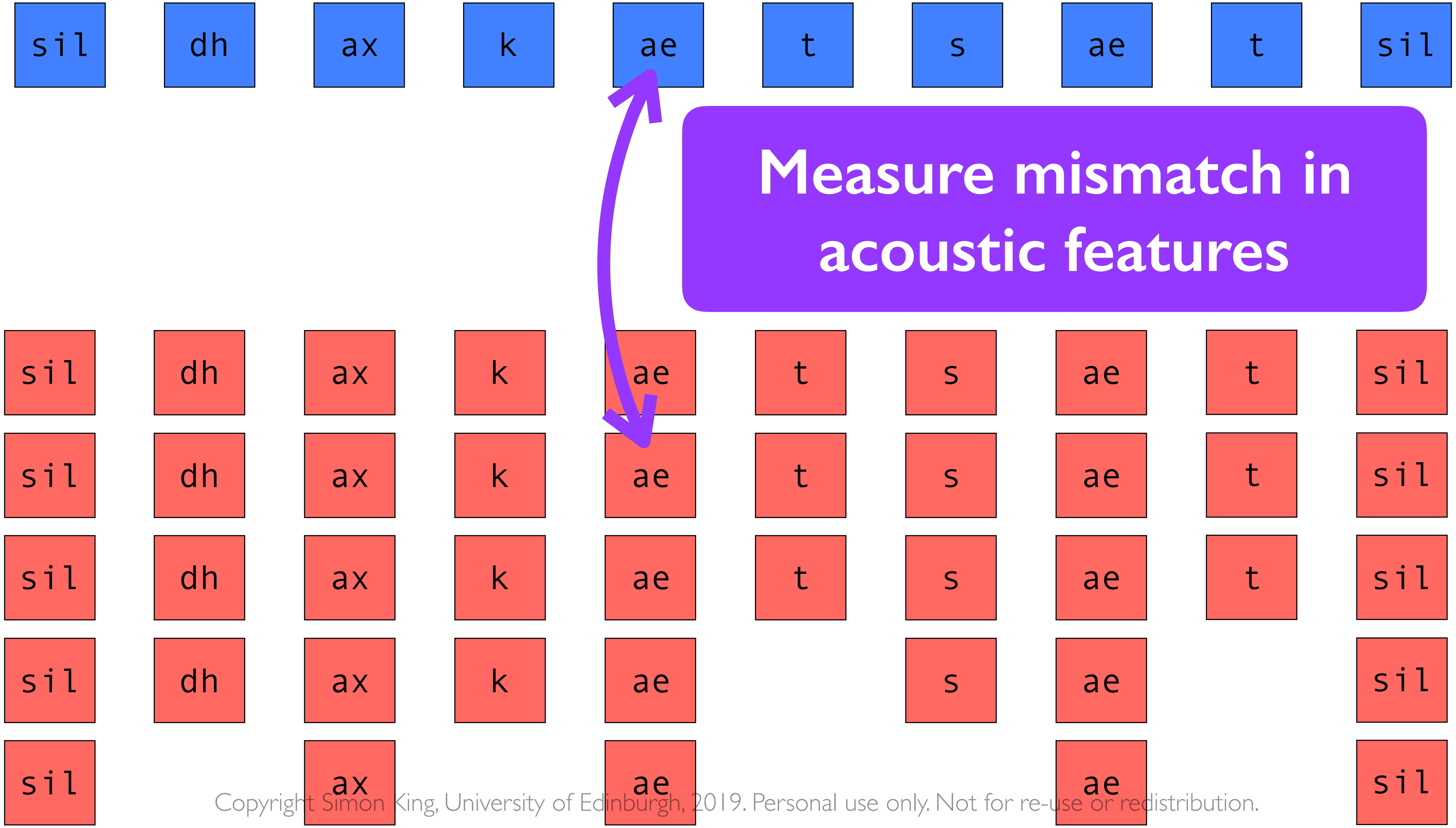
Regression

Predict acoustic features, given linguistic specification

sil	dh	ax	k	ae	t	s	ae	t	sil
sil	dh	ax	k	ae	t	s	ae	t	sil
sil	dh	ax	k	ae	t	s	ae	t	sil
sil	dh	ax	k	ae		s	ae		sil
sil		ax		ae			ae		sil

Unit selection using acoustic features

Waveform generator



IEEE Trans. Audio, Speech, and Language Proc. 21 (2), pp. 280-290, 2013. DOI:10.1109/TASL.2012.2221460

A Unified Trajectory Tiling Approach to High Quality Speech Rendering

Yao Qian, *Senior Member, IEEE*, Frank K. Soong, *Fellow, IEEE*, and Zhi-Jie Yan, *Member, IEEE*

***Abstract*—It is technically challenging to make a machine talk as naturally as a human so as to facilitate "frictionless" interactions between machine and human. We propose a trajectory tiling-based approach to high-quality speech rendering, where speech parameter trajectories, extracted from natural, processed, or synthesized speech, are used to guide the search for the best sequence of waveform "tiles" stored in a pre-recorded speech database. We test the proposed unified algorithm in both Text-To-Speech (TTS) syn-**

smooth and highly intelligible synthesized speech, it has still been perceived as a voice with some traditional vocoder flavor [10]. On the other hand, the waveform concatenation-based unit selection TTS can yield fairly natural sounding speech but occasionally it may still produce some undesirable concatenation glitches. The hybrid approaches, which use HMM to guide the unit selection process to minimize the spectral, pitch

Waveform generator

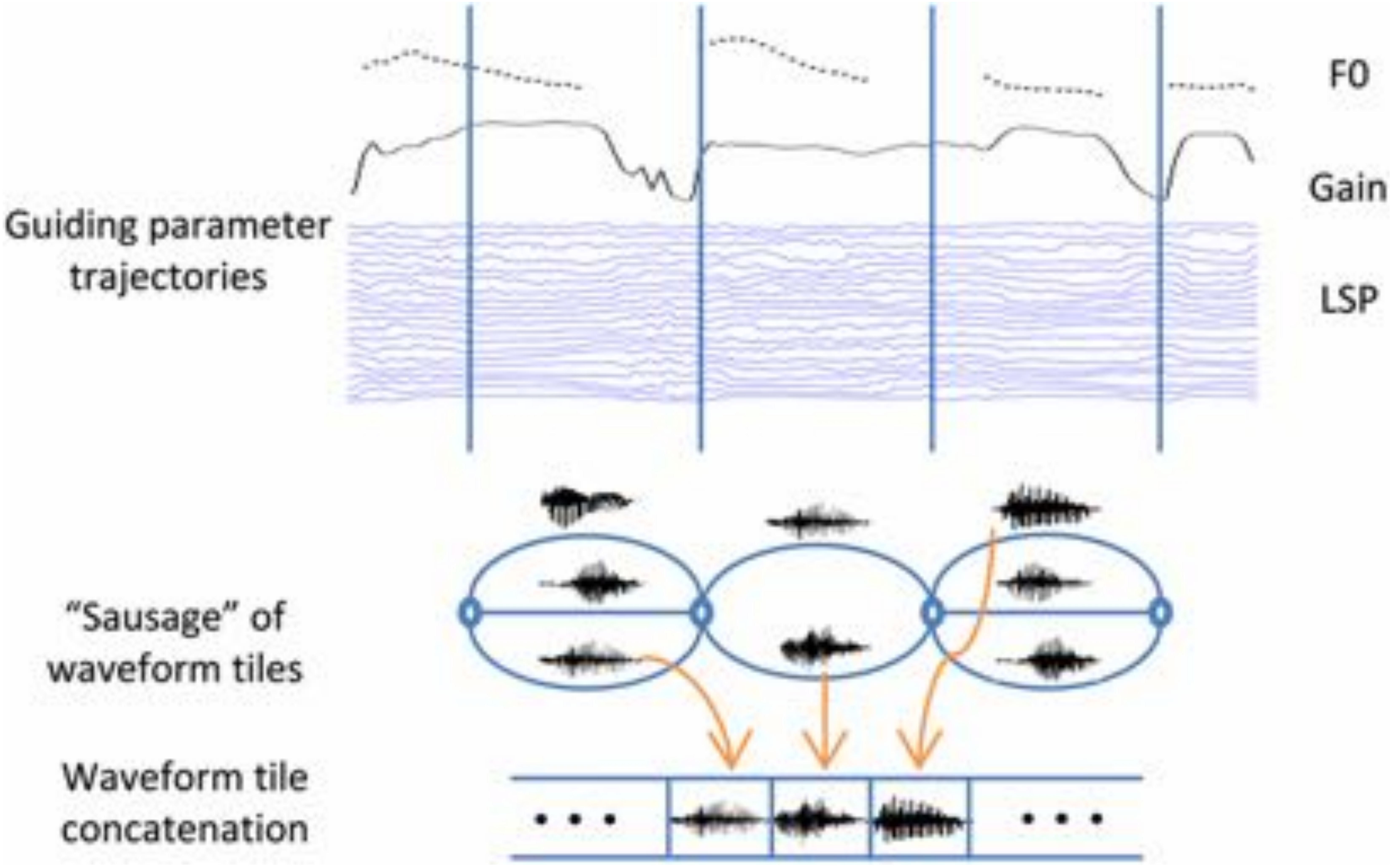


Figure 1 from Y. Qian, F. K. Soong and Z. J. Yan "A Unified Trajectory Tiling Approach to High Quality Speech Rendering" *IEEE Trans. Audio, Speech, and Language Proc.* 21 (2), pp. 280-290, 2013. DOI:10.1109/TASL.2012.2221460

Waveform generator

statistical parametric speech synthesis

neural speech synthesis

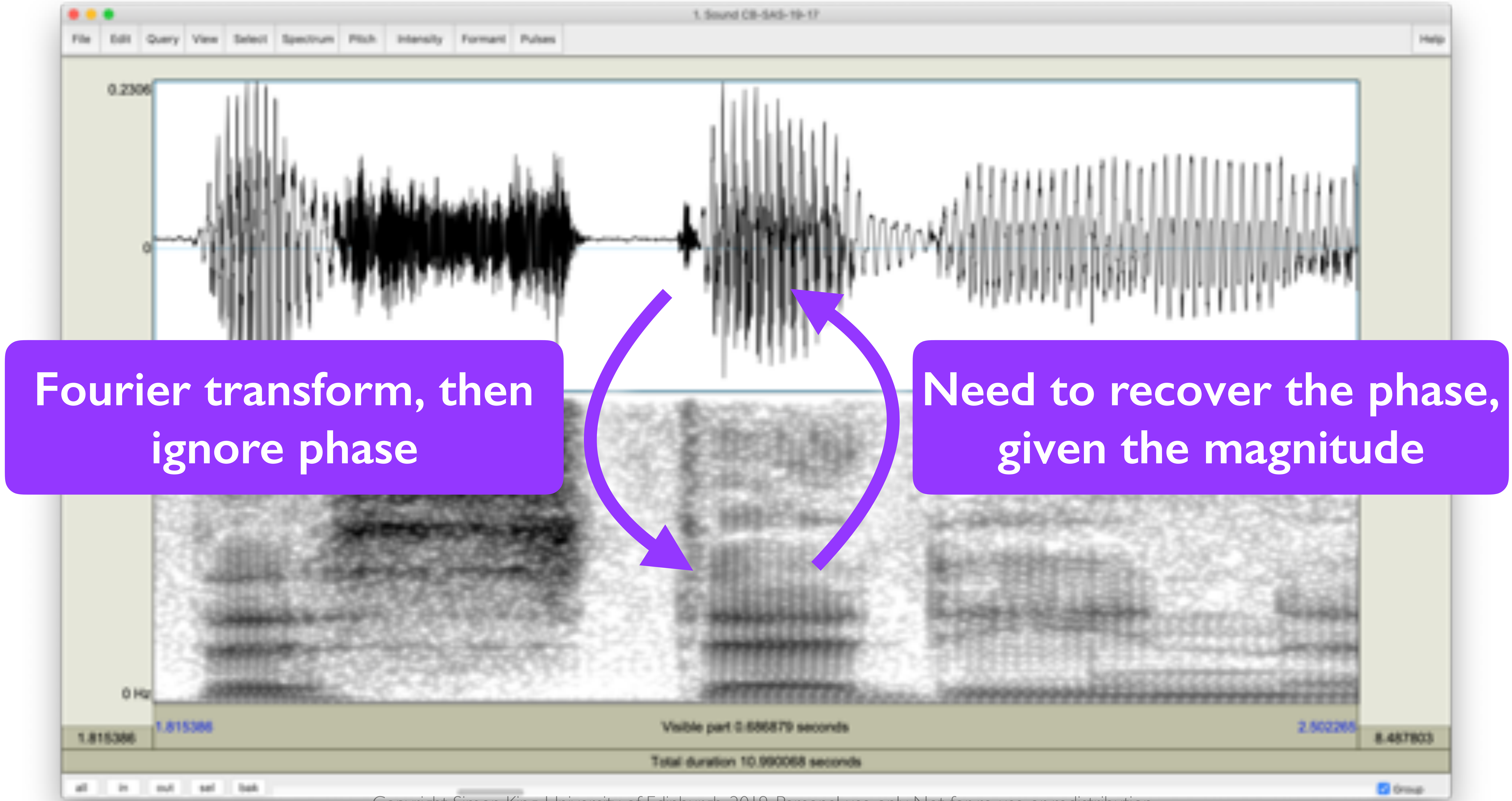
1st generation unit selection

2nd generation unit selection

1990 2000 2010 2020

Outline

- Tutorial
  - Text processing
  - Regression
  - Waveform generation
- Current research
  - <u>Waveform generation</u>
  - Regression
  - Text processing
- What next?


Fourier transform, then ignore phase
Need to recover the phase, given the magnitude

DOI: 10.21437/Interspeech.2019-1424 — Amazon's neural vocoder

INTERSPEECH 2019
September 15–19, 2019, Graz, Austria

Towards achieving robust universal neural vocoding

Jaime Lorenzo-Trueba[1], *Thomas Drugman*[1], *Javier Latorre*[1*], *Thomas Merritt*[1], *Bartosz Putrycz*[1], *Roberto Barra-Chicote*[1], *Alexis Moinet*[1], *Vatsal Aggarwal*[1]

[1]Amazon.com, Cambridge, United Kingdom

{truebaj, drugman, jlatorre, thommer, bartosz, rchicote, amoinet, agvatsal}@amazon.com

Abstract

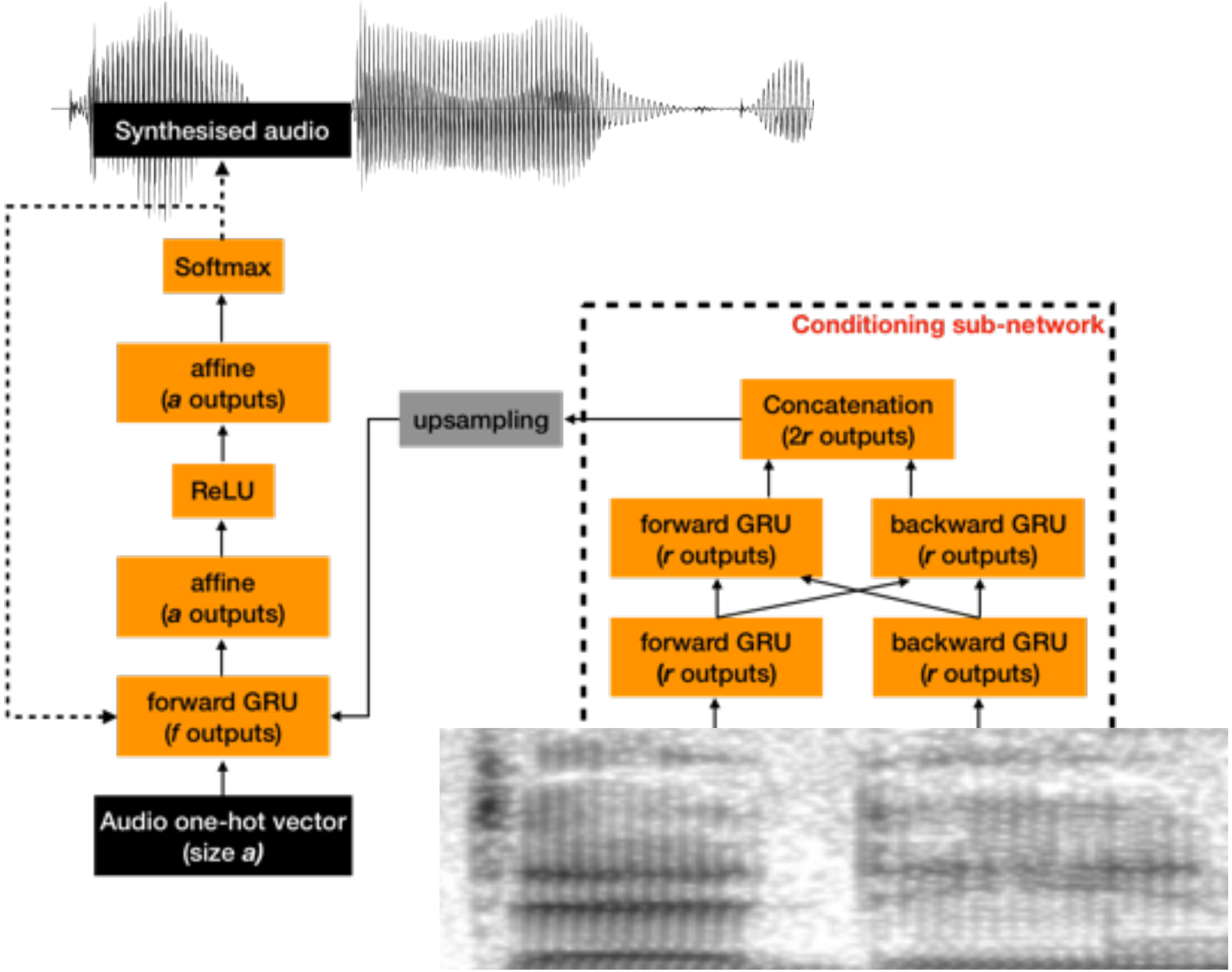
This paper explores the potential universality of neural vocoders. We train a WaveRNN-based vocoder on 74 speakers coming from 17 languages. This vocoder is shown to be capable of generating speech of consistently good quality (98% relative mean MUSHRA when compared to natural speech) re-

characteristics and have poor generalization capabilities [17]. Several recent studies attempted to improve the adaptation capabilities of such models [18, 19], commonly using explicit speaker information (either as a onehot encoding or some other form of speaker embedding) [20]. There are however reports in literature of initial successes training neural vocoders without providing explicit speaker information [21, 22], however the

Recap: doing regression with a neural network

Regression

Neural vocoder

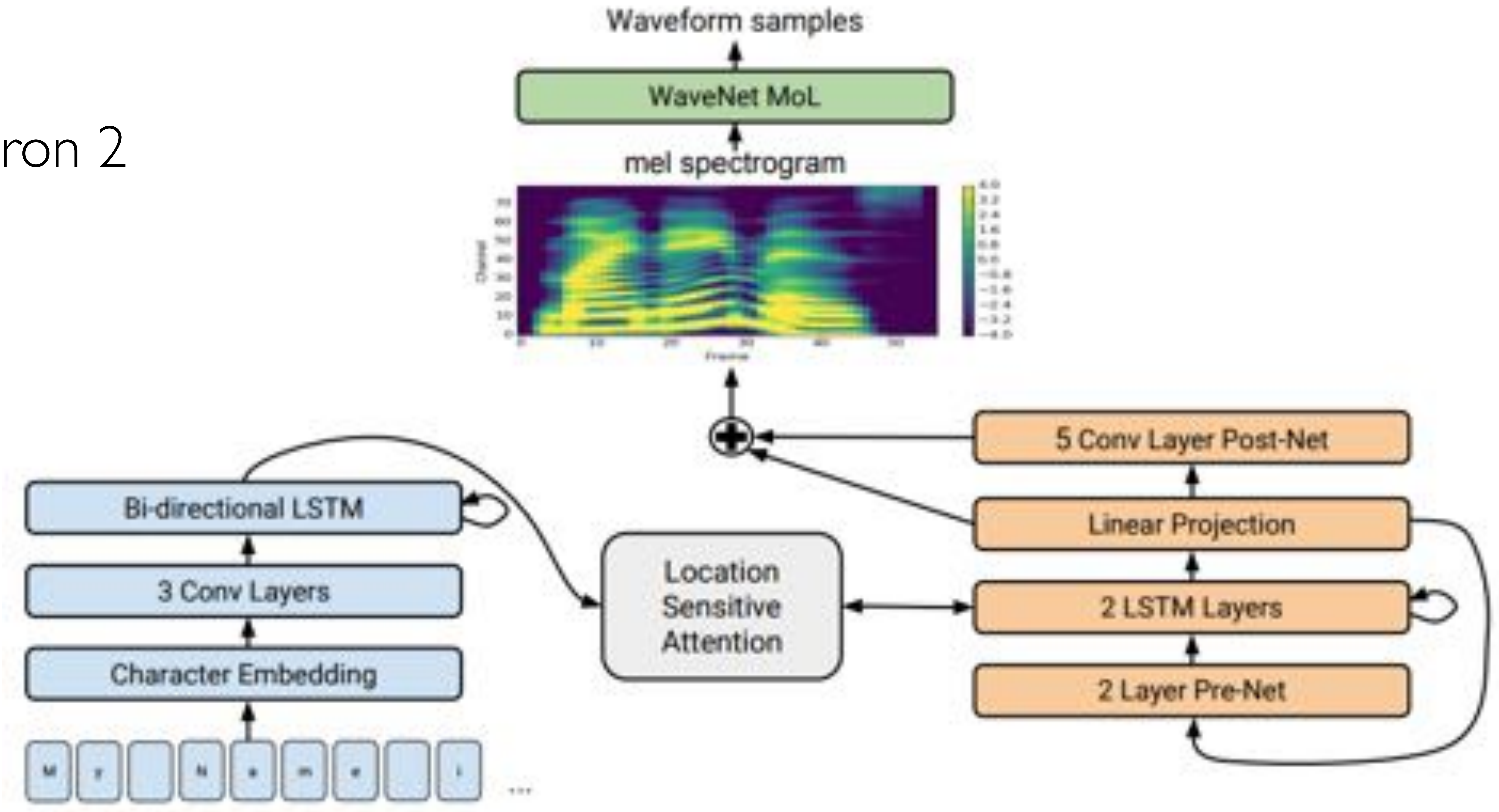


2 audio samples from an open-source implementation of Amazon's neural vocoder: https://bshall.github.io/UniversalVocoding/

Outline

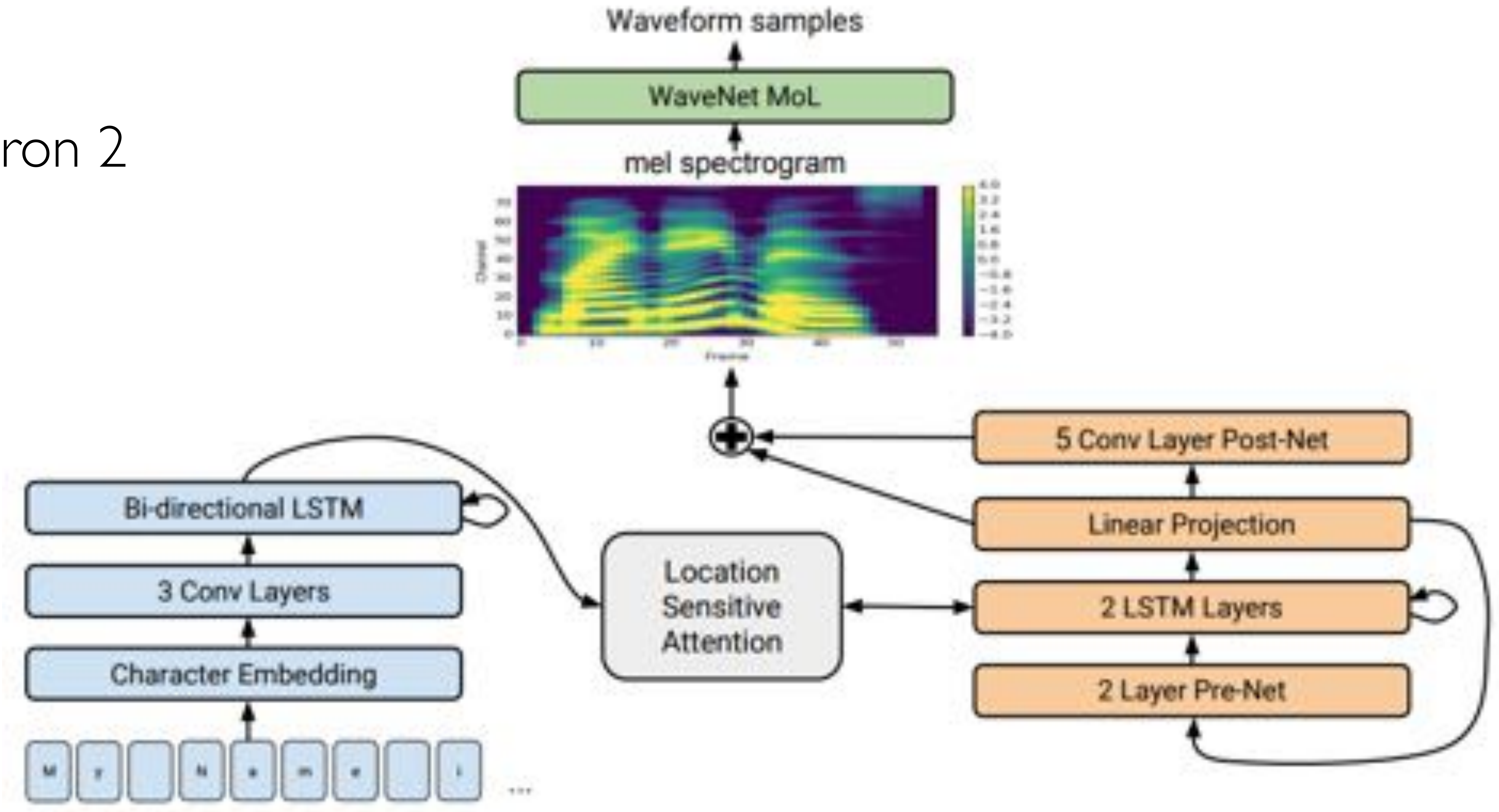
- Tutorial
  - Text processing
  - Regression
  - Waveform generation
- Current research
  - Waveform generation
  - <u>Regression</u>
  - Text processing
- What next?

Tacotron 2

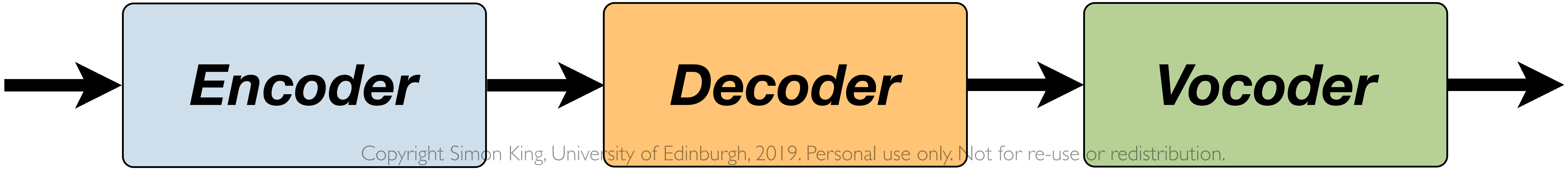

Waveform samples
WaveNet MoL
mel spectrogram
5 Conv Layer Post-Net
Linear Projection
2 LSTM Layers
2 Layer Pre-Net
Location Sensitive Attention
Bi-directional LSTM
3 Conv Layers
Character Embedding


Front end
Regression
Waveform generator

Tacotron 2

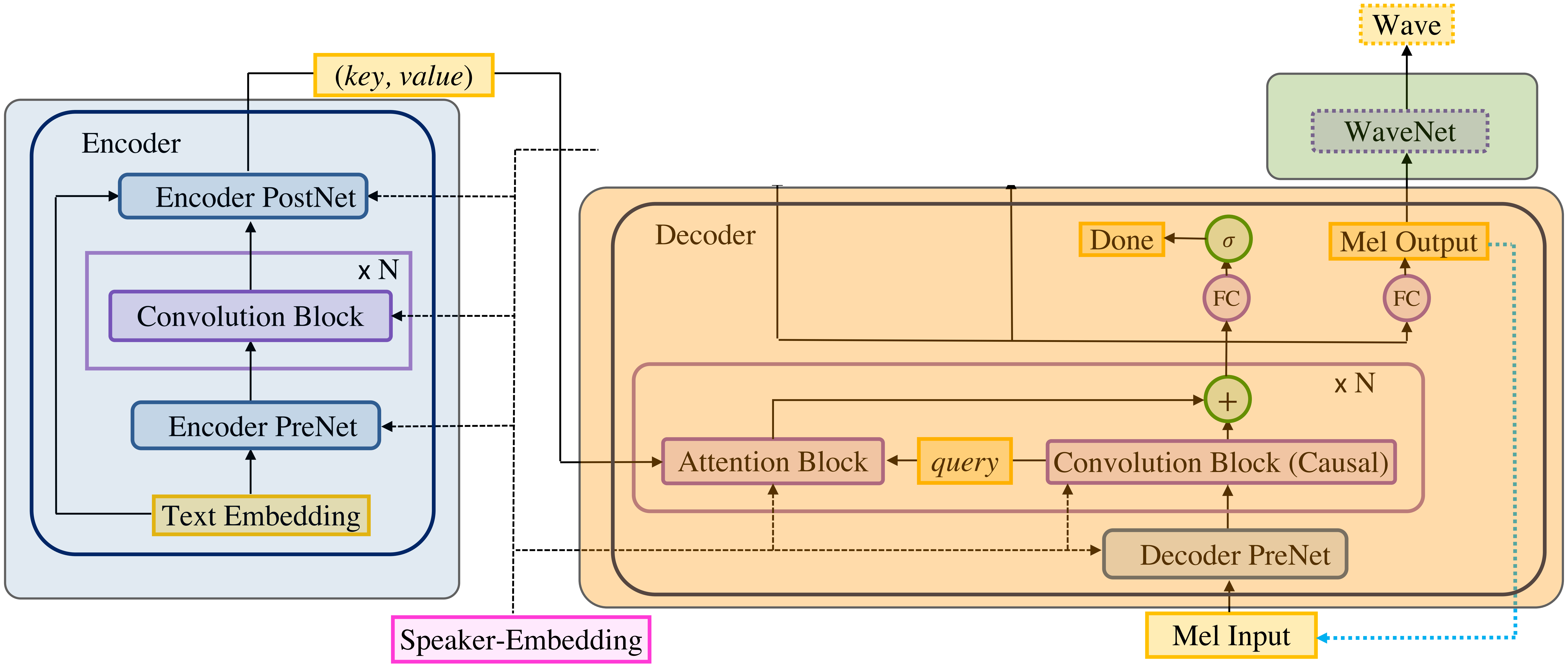


Waveform samples
WaveNet MoL
mel spectrogram
5 Conv Layer Post-Net
Linear Projection
2 LSTM Layers
2 Layer Pre-Net
Location Sensitive Attention
Bi-directional LSTM
3 Conv Layers
Character Embedding

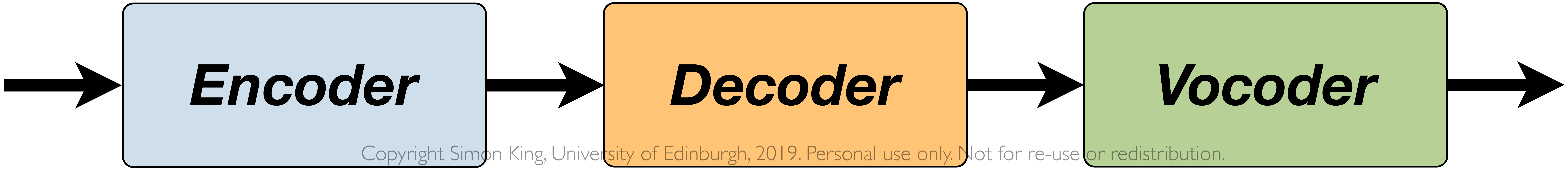


Encoder
Decoder
Vocoder

Deep Voice 3



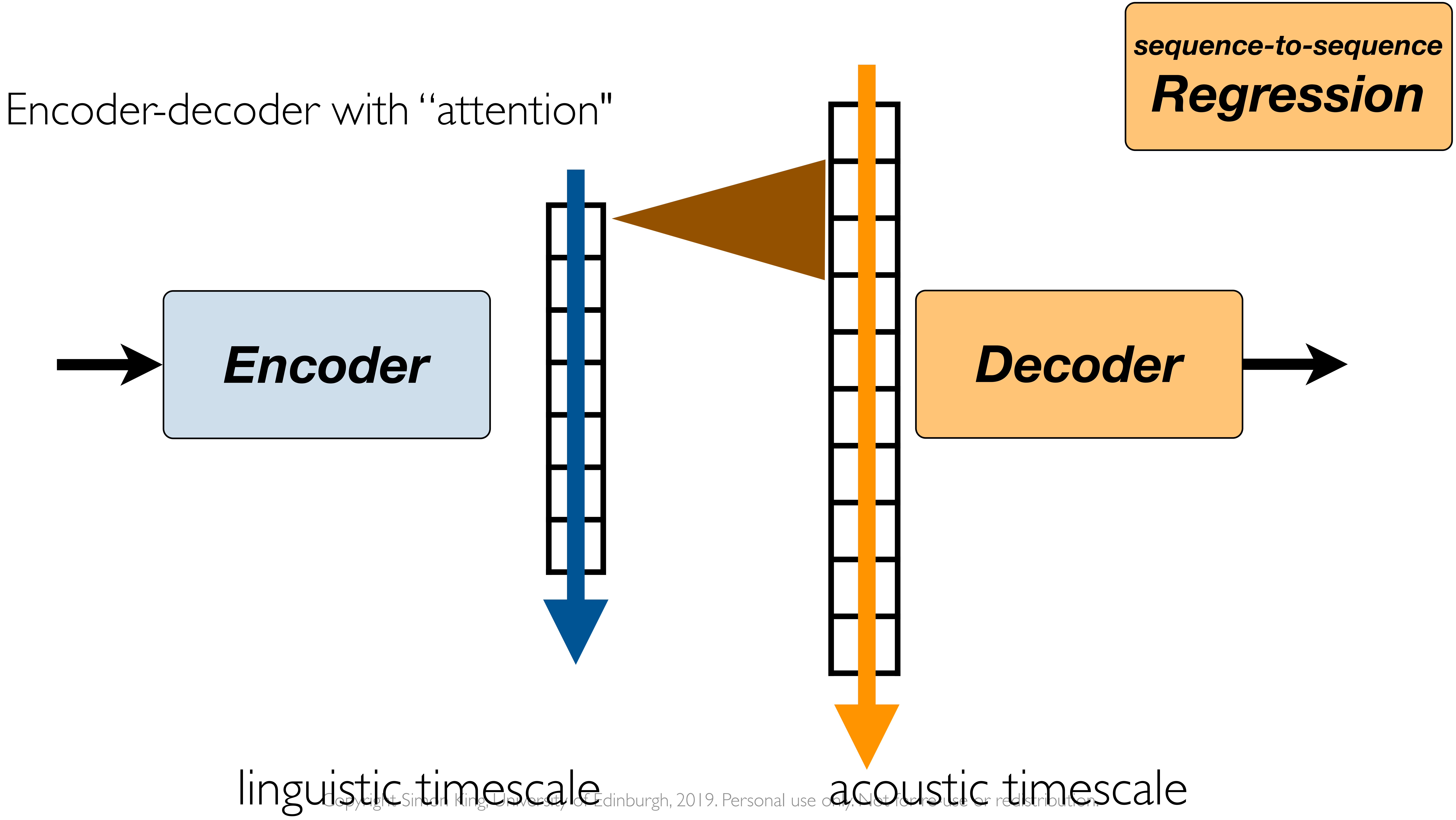
(key, value)
Encoder
Encoder PostNet
x N
Convolution Block
Encoder PreNet
Text Embedding
Speaker-Embedding
Wave
WaveNet
Decoder
Done
σ
Mel Output
FC
FC
x N
+
Attention Block
query
Convolution Block (Causal)
Decoder PreNet
Mel Input



Encoder
Decoder
Vocoder

Encoder-decoder

sequence-to-sequence
Regression


sequence-to-sequence
Regression
Encoder-decoder with "attention"
Encoder
Decoder
linguistic timescale
acoustic timescale

Style Tokens: Unsupervised Style Modeling, Control and Transfer in End-to-End Speech Synthesis

Yuxuan Wang [1] Daisy Stanton [1] Yu Zhang [1] RJ Skerry-Ryan [1] Eric Battenberg [1] Joel Shor [1] Ying Xiao [1]
Fei Ren [1] Ye Jia [1] Rif A. Saurous [1]

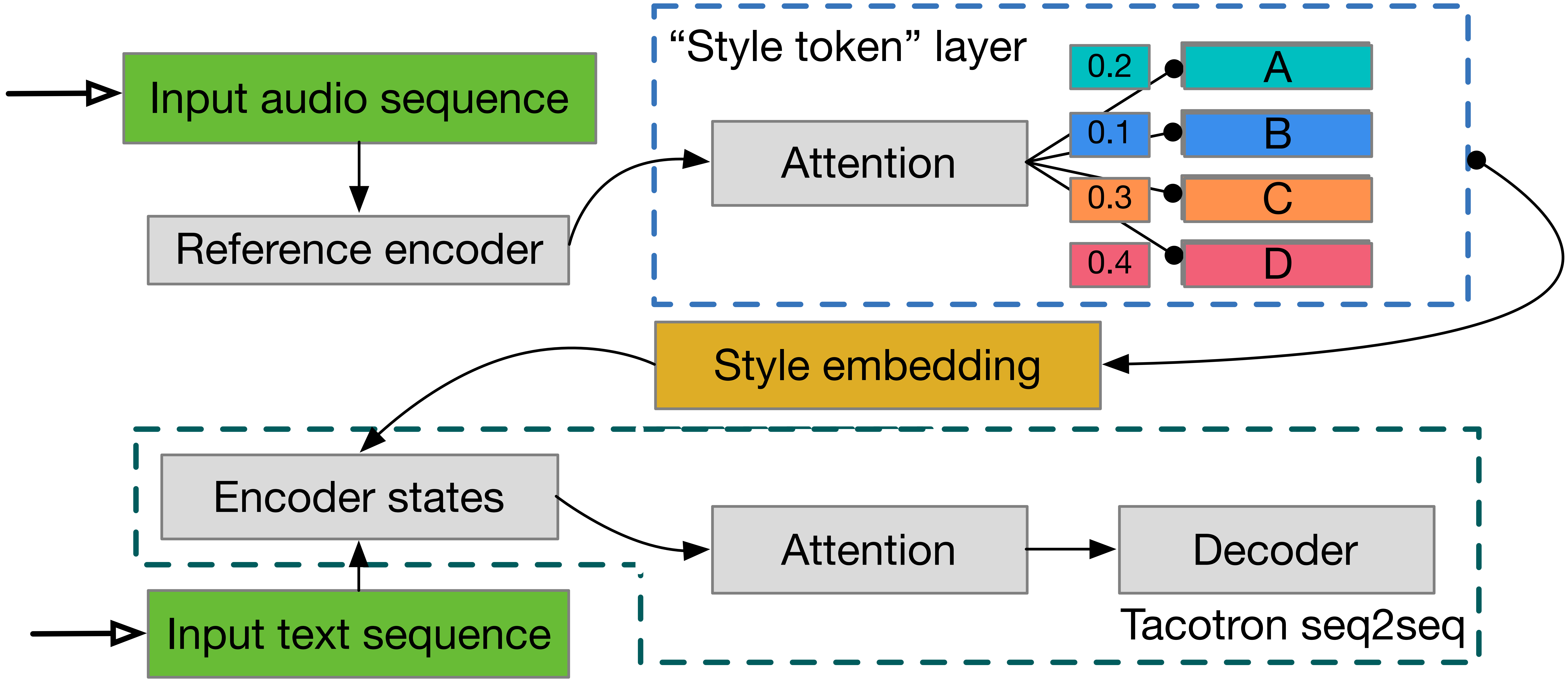
23 Mar 2018

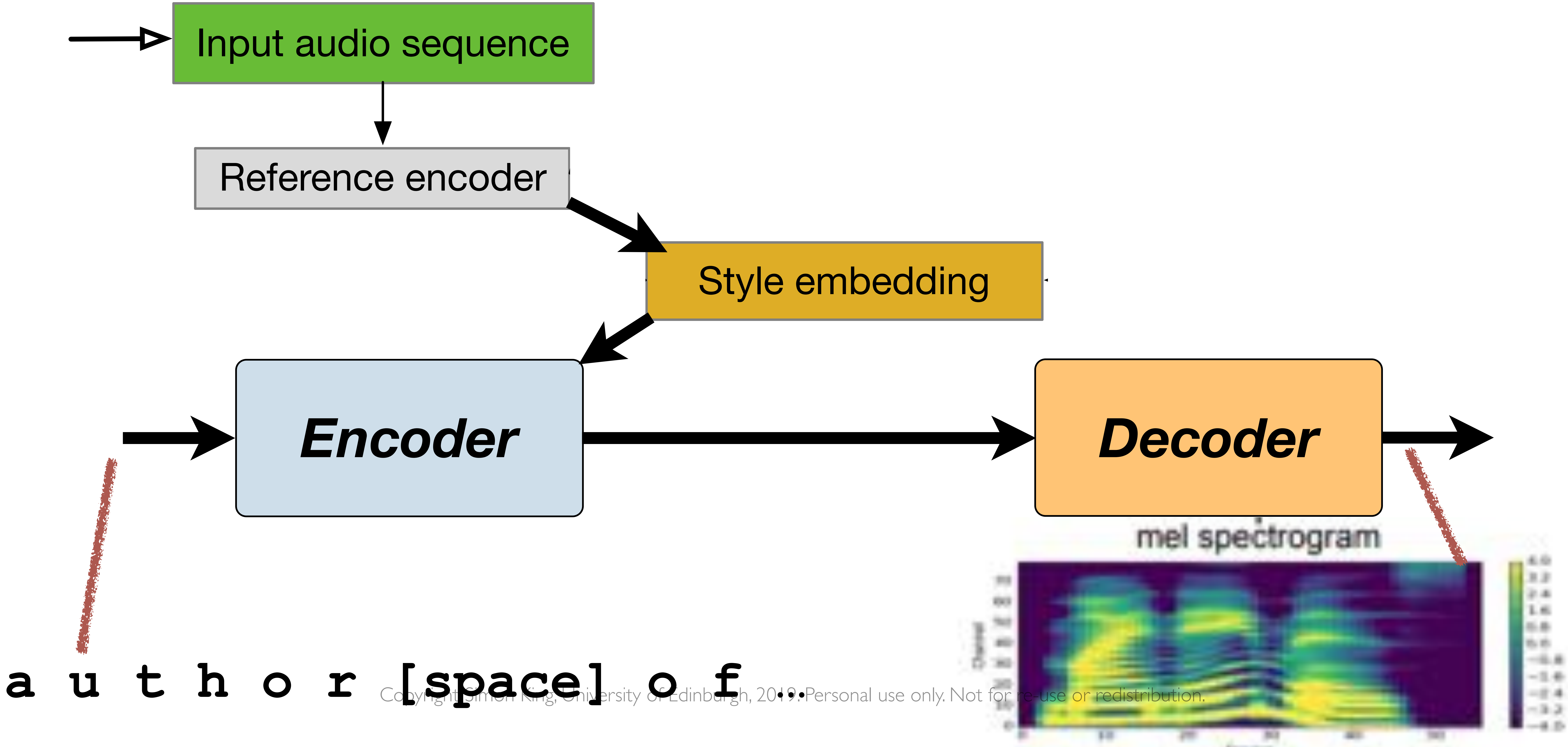
Abstract

In this work, we propose "global style tokens" (GSTs), a bank of embeddings that are jointly trained within Tacotron, a state-of-the-art end-to-end speech synthesis system. The embeddings are trained with no explicit labels, yet learn to model a large range of acoustic expressiveness. GSTs lead to a rich set of significant results. The soft interpretable "labels" they generate can be

on *style modeling*, the goal of which is to provide models the capability to choose a speaking style appropriate for the given context. While difficult to define precisely, style contains rich information, such as intention and emotion, and influences the speaker's choice of intonation and flow. Proper stylistic rendering affects overall perception (see e.g. "affective prosody" in (Taylor, 2009)), which is important for applications such as audiobooks and newsreaders.

Style modeling presents several challenges. First, there is no


Input audio sequence
Reference encoder
"Style token" layer
Attention
0.2
0.1
0.3
0.4
A
B
C
D
Style embedding
Encoder states
Attention
Decoder
Tacotron seq2seq
Input text sequence


Input audio sequence
Reference encoder
Style embedding
Encoder
Decoder
mel spectrogram
a u t h o r [space] o f …

What is an "embedding" ?

How do you learn an "embedding" ?

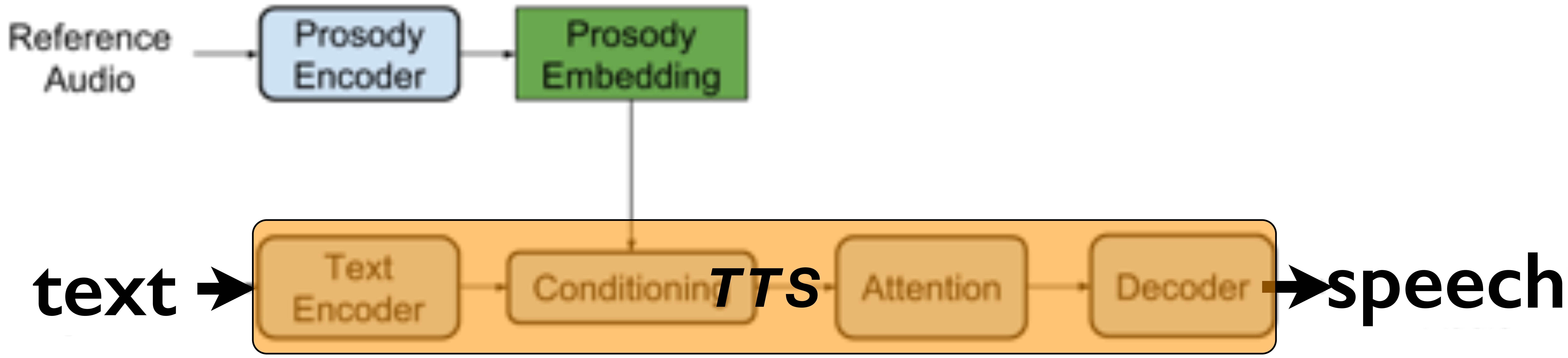


Figure from https://ai.googleblog.com/2018/03/expressive-speech-synthesis-with.html

How do you learn an "embedding"?

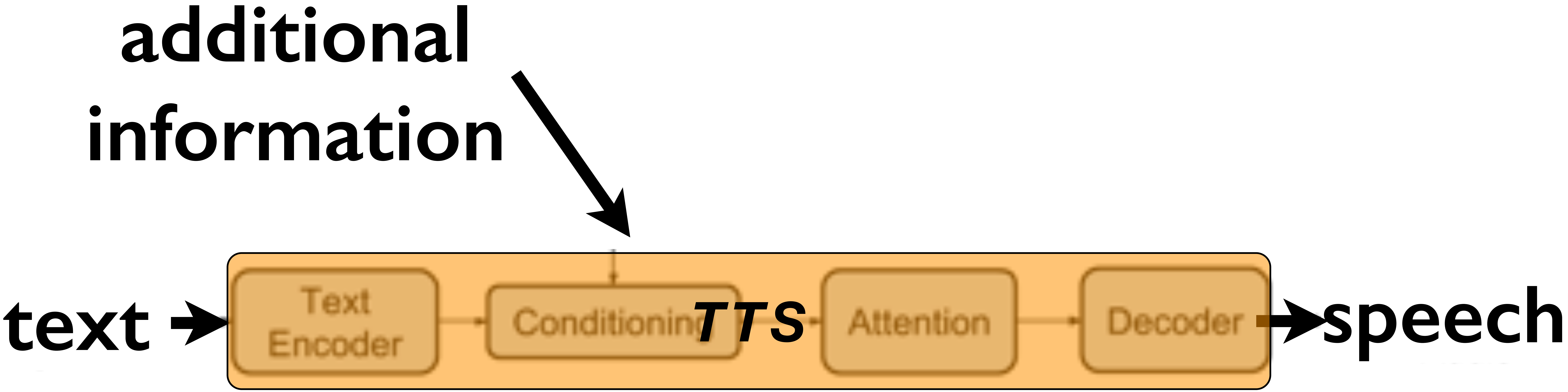
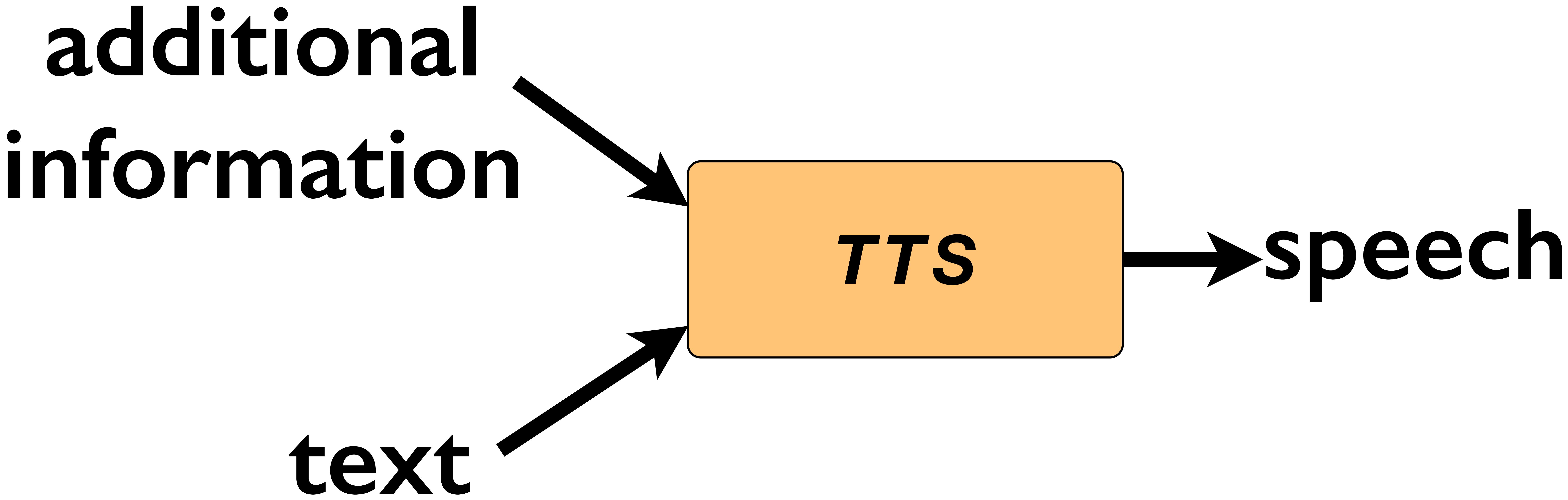
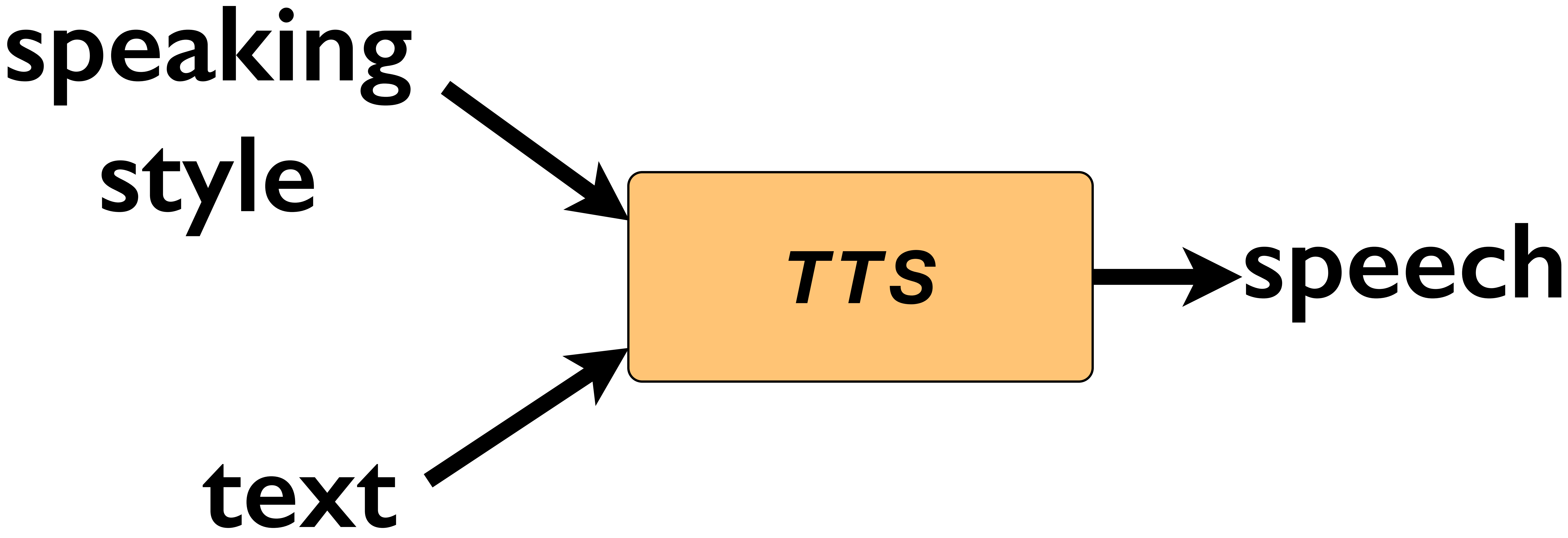


Figure from https://ai.googleblog.com/2018/03/expressive-speech-synthesis-with.html

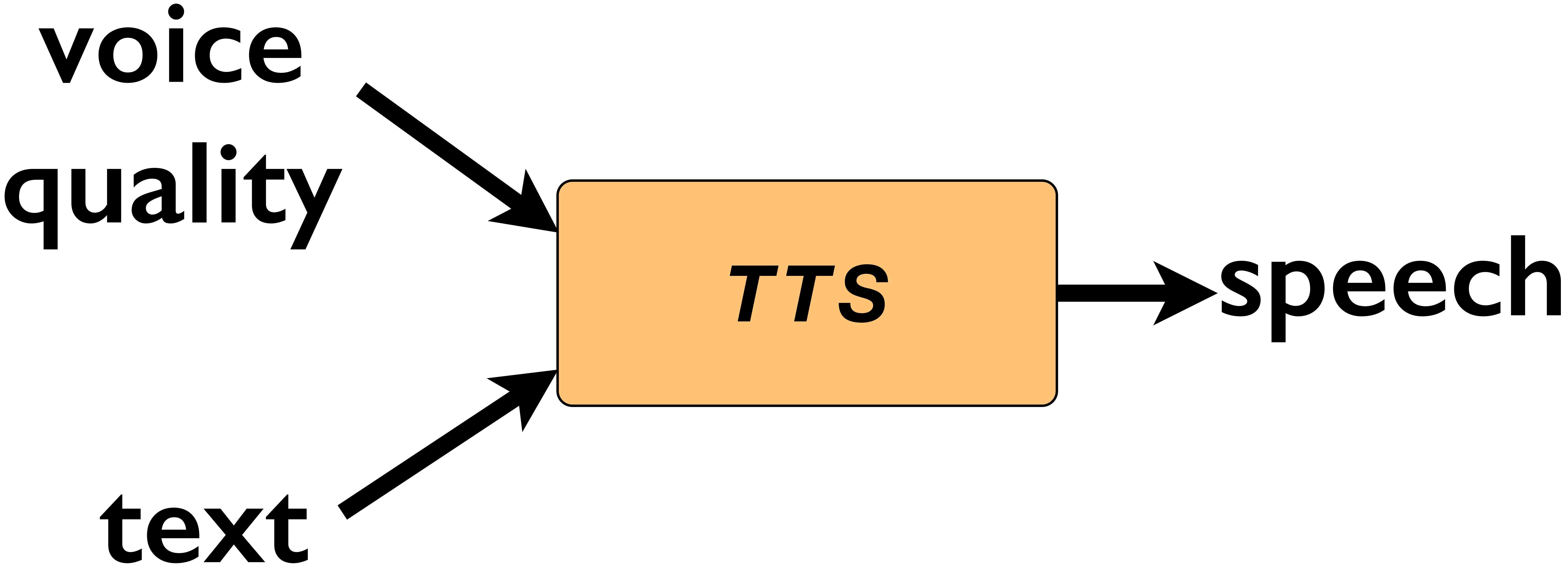
The **text** is not enough to entirely ***explain*** the **speech**



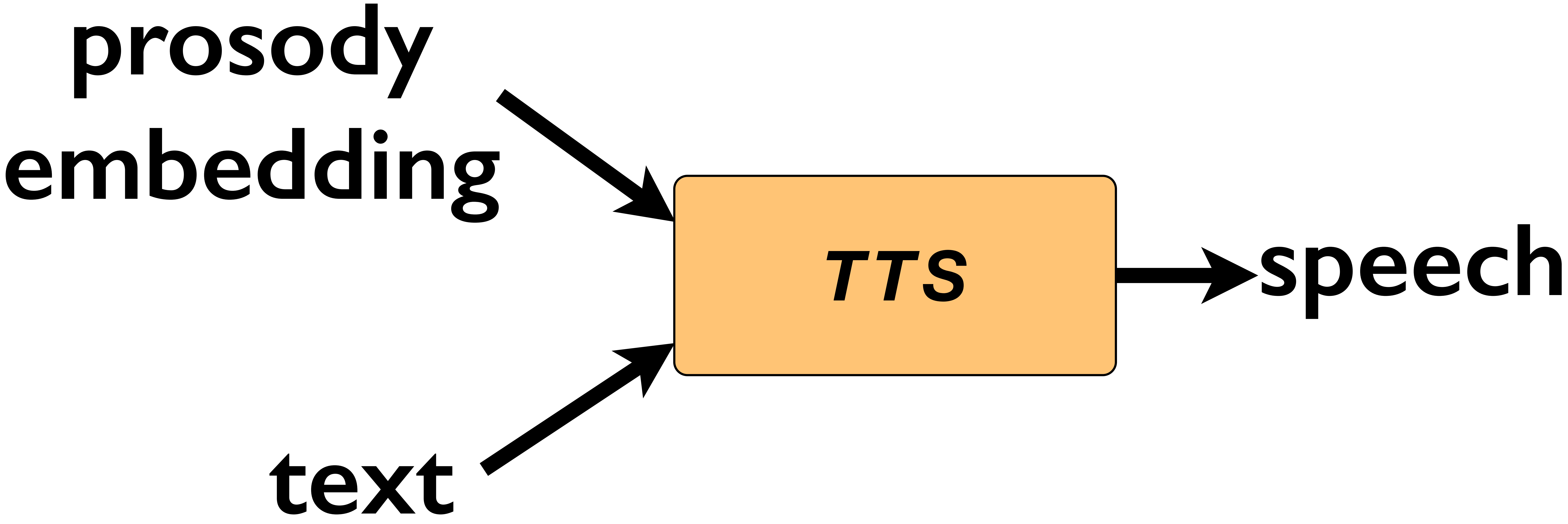
The **text** is not enough to entirely ***explain*** the **speech**



The **text** is not enough to entirely ***explain*** the **speech**



Additional information derived from a **reference audio sample**

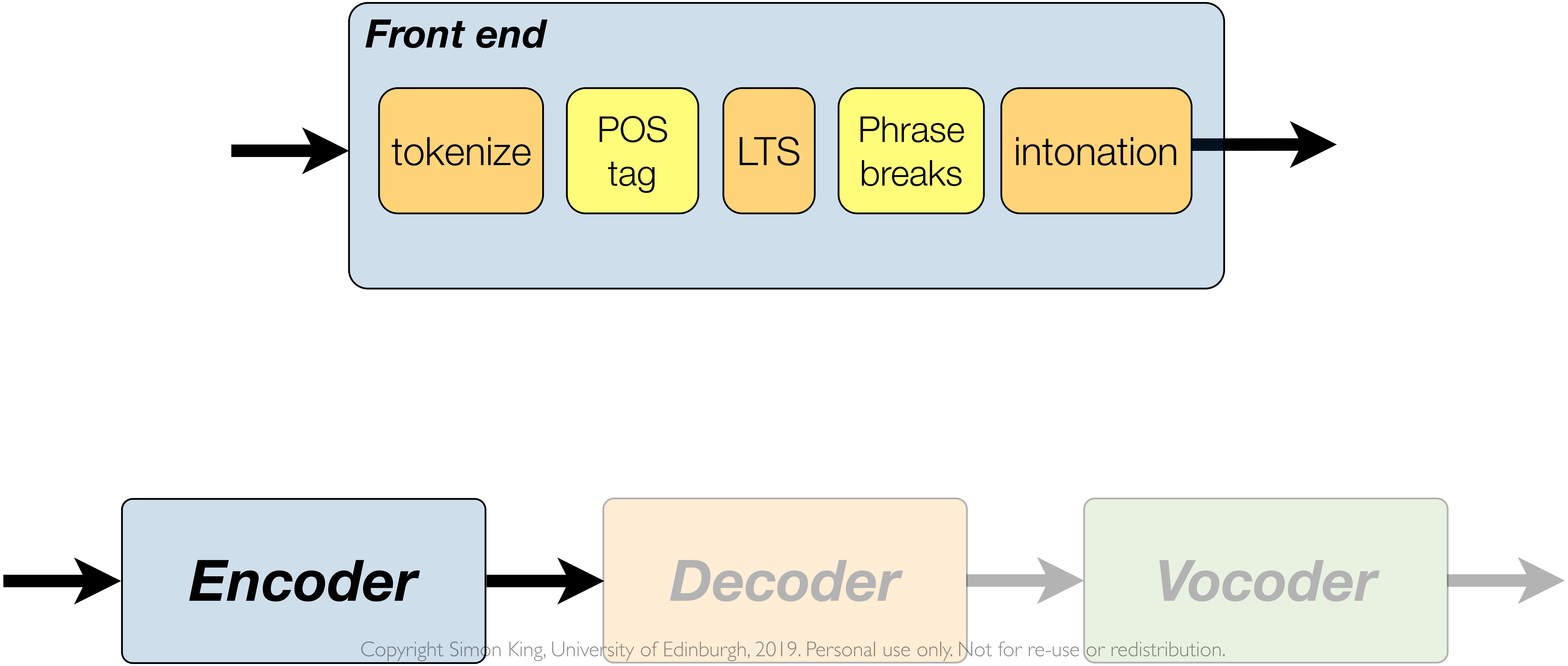


5 audio samples from https://google.github.io/tacotron/publications/global_style_tokens

Outline

- Tutorial
  - Text processing
  - Regression
  - Waveform generation
- Current research
  - Waveform generation
  - Regression
  - <u>Text processing</u>
- What next?

Traditional vs New



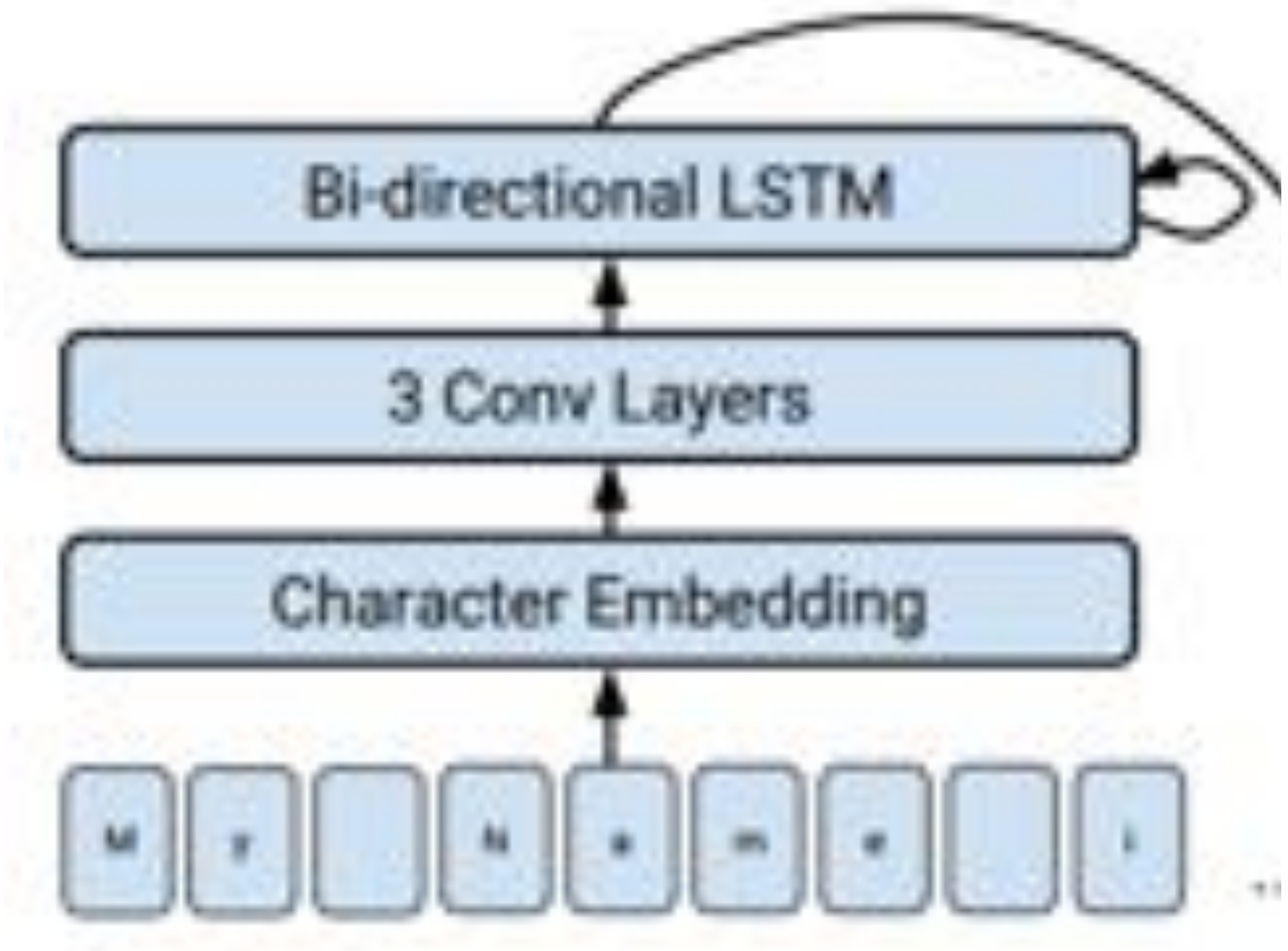
Traditional — explicit pronunciation dictionary + letter-to-sound model

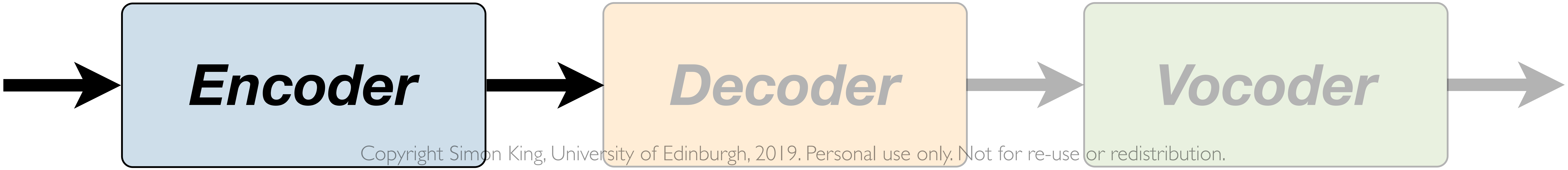
from 20k up to 200k entries (unique types)

```
AERIALS   EH1 R IY0 AH0 L Z
AERIE   EH1 R IY0
AERIEN   EH1 R IY0 AH0 N
AERIENS   EH1 R IY0 AH0 N Z
AERITALIA   EH2 R IH0 T AE1 L Y AH0
AERO   EH1 R OW0
AEROBATIC   EH2 R AH0 B AE1 T IH0 K
AEROBATICS   EH2 R AH0 B AE1 T IH0 K S
AEROBIC   EH0 R OW1 B IH0 K
AEROBICALLY   EH0 R OW1 B IH0 K L IY0
AEROBICS   ER0 OW1 B IH0 K S
AERODROME   EH1 R AH0 D R OW2 M
AERODROMES   EH1 R AH0 D R OW2 M Z
AERODYNAMIC   EH2 R OW0 D AY0 N AE1 M IH0 K
AERODYNAMICALLY   EH2 R OW0 D AY0 N AE1 M IH0 K L IY0
AERODYNAMICIST   EH2 R OW0 D AY0 N AE1 M IH0 S IH0 S T
AERODYNAMICISTS   EH2 R OW0 D AY0 N AE1 M IH0 S IH0 S T S
AERODYNAMICISTS(1)   EH2 R OW0 D AY0 N AE1 M IH0 S IH0 S
AERODYNAMICS   EH2 R OW0 D AY0 N AE1 M IH0 K S
AERODYNE   EH1 R AH0 D AY2 N
AERODYNE'S   EH1 R AH0 D AY2 N Z
AEROFLOT   EH1 R OW0 F L AA2 T
```

+ a statistical model learned from this data

New — encoder learns a character sequence embedding





Tacotron 2

While our samples sound great, there are still some difficult problems to be tackled. For example, our system has difficulties pronouncing complex words (such as "decorum" and "merlot"),

from https://ai.googleblog.com/2017/12/tacotron-2-generating-human-like-speech.html

- Are "decorum" and "merlot" *really* **complex** words?
- The Oxford British English dictionary says

DECORUM	dɪˈkɔːrəm
MERLOT	ˈməːləʊ/

- Which doesn't seem *particularly* difficult …

Submitted to ICASSP 2020

ENHANCING SEQUENCE-TO-SEQUENCE TEXT-TO-SPEECH WITH MORPHOLOGY

*Jason Taylor** *and Korin Richmond*†

Centre for Speech Technology Research, The University of Edinburgh, UK

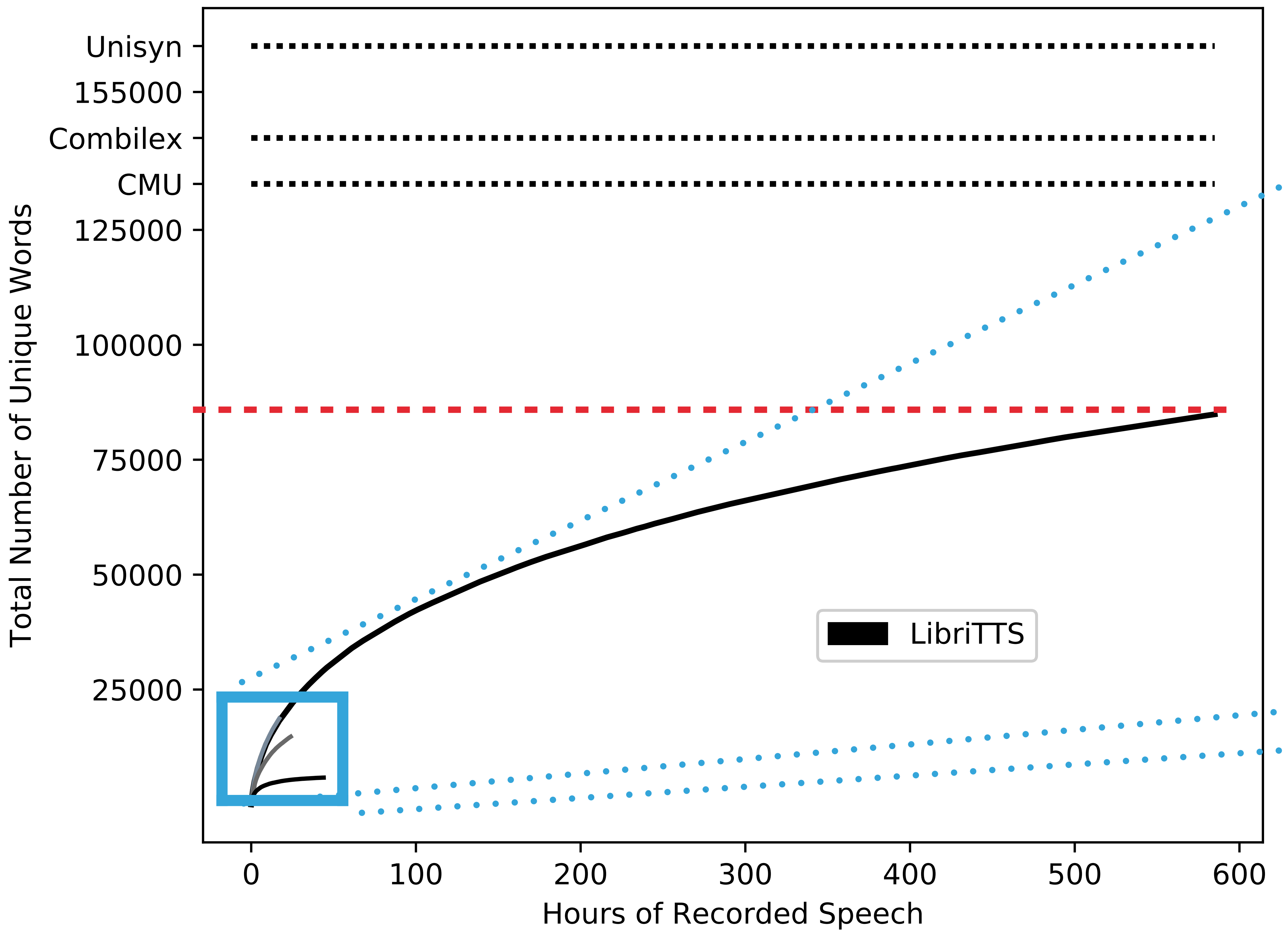
ABSTRACT

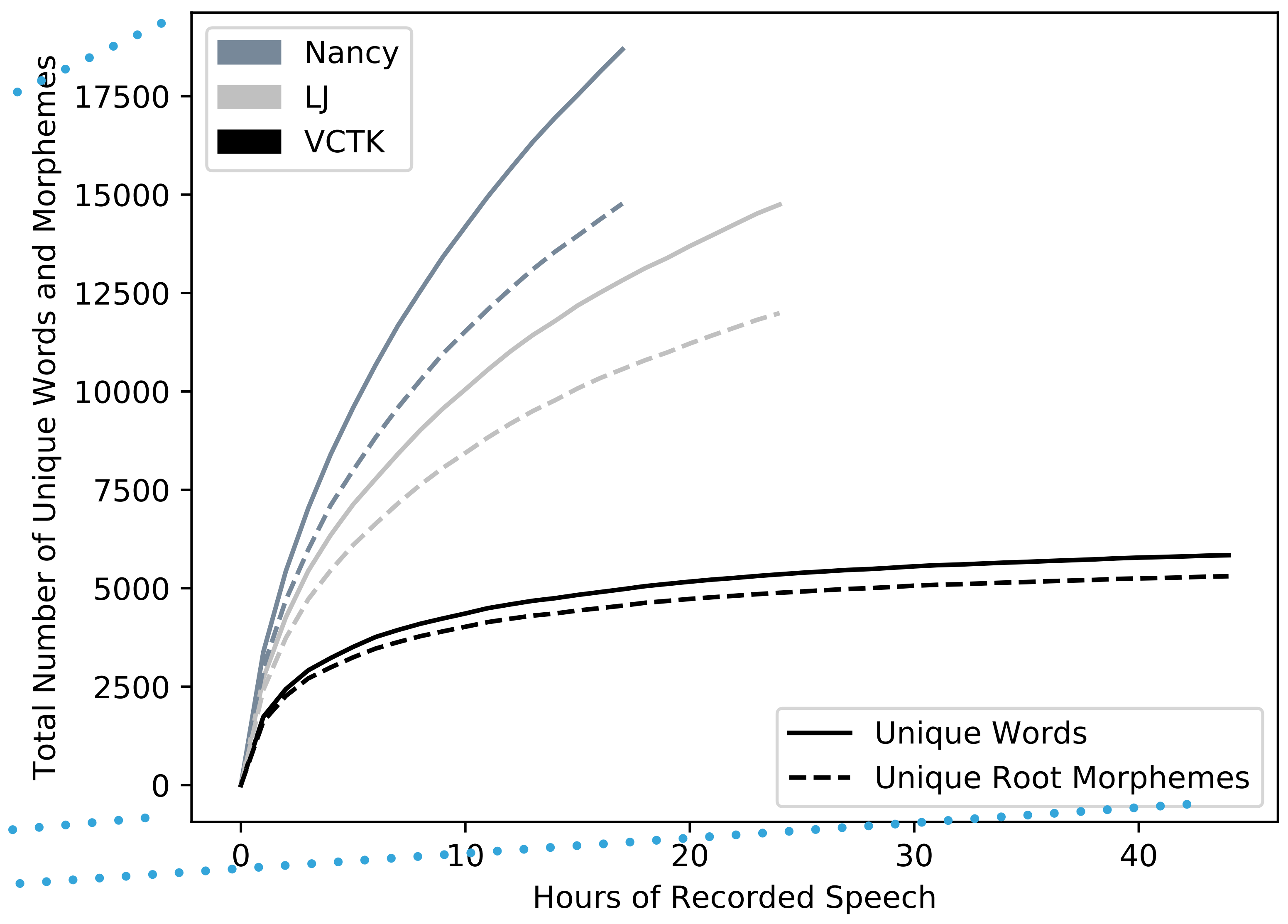
Neural sequence-to-sequence (S2S) modelling encodes a single, unified representation for each input sequence. In the field of text-to-speech (TTS), such representations embed ambiguities between English spelling and pronunciation. For example, in *pothole* and *there* the character cluster *th* sounds different. This is problematic when predicting pronunciation directly from letters. When letters are grouped into sub-word units like morphemes, we posit pronunciation will become easier to predict. Accordingly, we test the effect of augmenting input sequences of letters with morphological boundaries. We find morphological boundaries substantially lower

front-end packages include Festival [4], Mary [5] and Sparrowhawk [6]. Front-end modules are limited in their coverage and rely on a back-off G2P model. This means improving pronunciation prediction from letters is still valuable in TTS.

Neural Sequence-to-Sequence (S2S) models are the current state-of-the-art in both TTS [7] and G2P modelling [8]. Both tasks involve the prediction of pronunciation from letters, either implicitly or explicitly. By implicitly, we mean the pronunciation is learnt latently and only inferred from output audio, as in end-to-end (E2E) TTS systems such as Tacotron [9]. By explicitly, we refer to the explicit prediction of phones, as in G2P. The vagaries of English spelling make pronunciation prediction by S2S models error-prone [10].

Datasets have low lexical coverage





Morphology should help

- Morphological boundaries break up words into constituent parts:
  - **coathanger** is {coat}{hang}>er>
- Can disambiguate pronunciation ambiguities (**th** in the above example)

Input	Base Unit	Morphs	Format	V
G	Graphemes	✗	p o t h o l e s	13981
GM	Graphemes	✓	{ p o t } { h o l e } >s >	5202
P	Phones	✗	p o t h ou l z	12631
PM	Phones	✓	{ p o t } {h ou l } >z >	5606

Naturalness of TTS using various forms of input

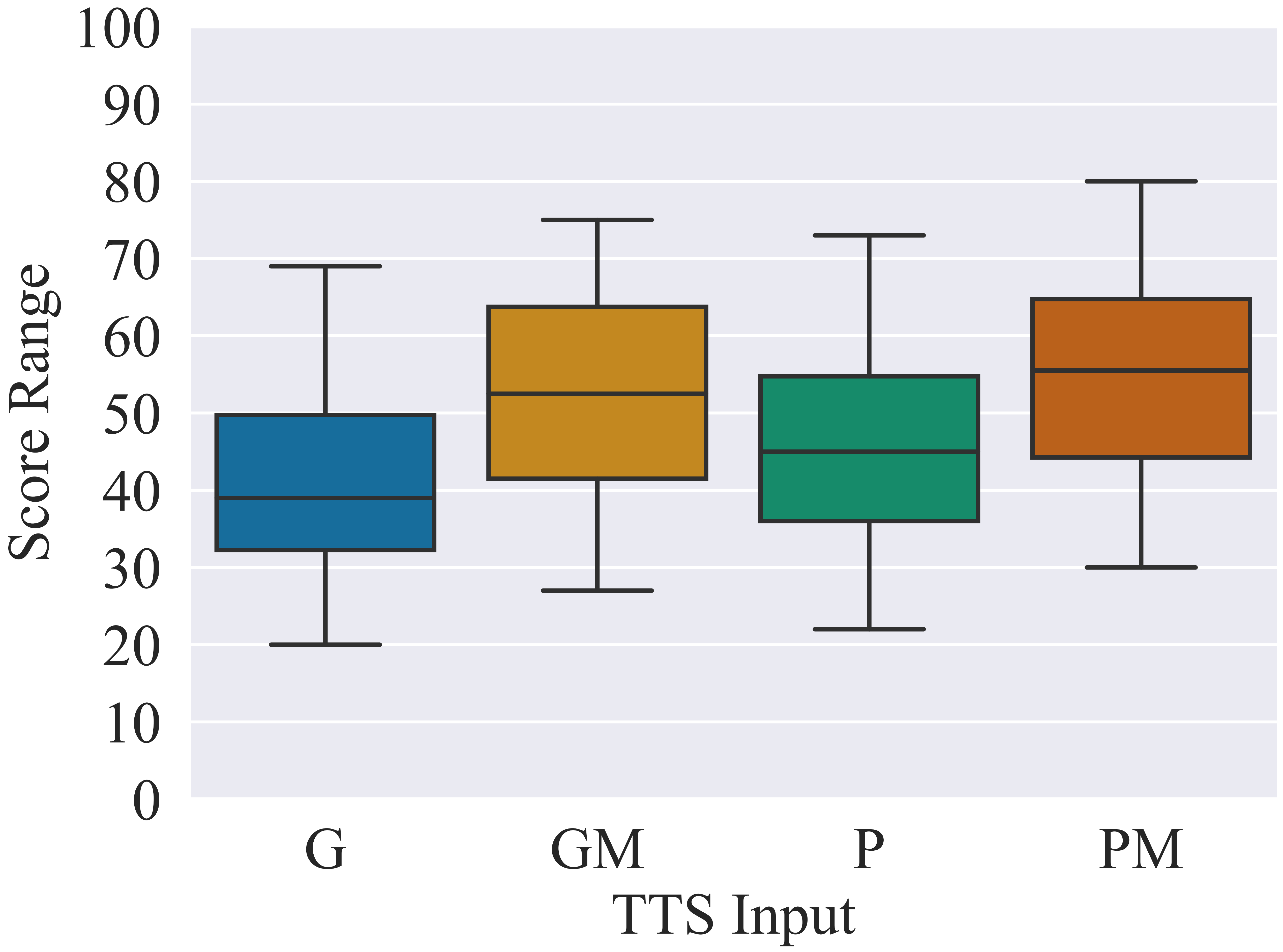
G = graphemes

P = phonemes

M = with morphology

Regression: Tacotron

Waveform generation: neural vocoder



Morphological boundaries improve pronunciation learning

G input	GM input	G Pronunciation (Incorrect)	GM Pronunciation (Correct)
coathanger	{coat}{hang}>er>	[kʌθˈəɪndʒə]	[koʊtˈhæŋə]
pothole	{pot}{hole}	[pɒˈθəl]	[pɒtˈhɒl]
goatherd	{goat}{herd}	[ˈgɒðəd]	[ˈgəʊtheɹd]
loophole	{loop}{hole}	[luˈfɒl]	[ˈluphəʊl]
upheld	{up}{held}	[ˈʌfɛld]	[ʌpˈhɛld]
cowherd	{cow}{herd}	[ˈkaʊɛɹd]	[kaʊˈhɛɹdˈ]
gigabytes	<giga<{byte}>s>	[gɪˈgaːbɪts]	[gɪgəˈbaɪts]
wobbliest	{wobble}>y>>est>	[ˈwɒblɪst]	[wɒˈbliɛst]
optimisers	{optim==ise}>er>>	[ˈoptɪmɪzəz]	[optɪˈmaɪzəz]
synchronizable	{syn==chron==ize}>able>	[ˌsɪˈʧɹaɪzəbɫ]	[ˌsiŋkreʊˈnaɪzəbɫ]

More samples: http://homepages.inf.ed.ac.uk/s1649890/morph/

Morphological boundaries improve pronunciation learning

G input	GM input	G Pronunciation (Incorrect)	GM Pronunciation (Correct)
coathanger	{coat}{hang}>er>	[kʌθˈəɪnʤə]	[koʊtˈhæŋə]
pothole	{pot}{hole}	[pɒˈθəl]	[pɒtˈhɒl]
goatherd	{goat}{herd}	[ˈgɒðəd]	[ˈgəʊtheɹd]
loophole	{loop}{hole}	[luˈfɒl]	[ˈluphəʊl]
upheld	{up}{held}	[ˈʌfɛld]	[ʌpˈhɛld]
cowherd	{cow}{herd}	[ˈkaʊɛɹd]	[kaʊˈhɛɹdˈ]
gigabytes	<giga<{byte}>s>	[gɪˈgaːbɪts]	[gɪgəˈbaɪts]
wobbliest	{wobble}>y>>est>	[ˈwɒblɪst]	[wɒˈbliɛst]
optimisers	{optim==ise}>er>>	[ˈoptɪmɪzəz]	[optɪˈmaɪzəz]
synchronizable	{syn==chron==ize}>able>	[ˌsɪˈʧɹaɪzəbɫ]	[ˌsiŋkreʊˈnaɪzəbɫ]

coathanger

More samples: http://homepages.inf.ed.ac.uk/s1649890/morph/

Morphological boundaries improve pronunciation learning

G input	GM input	G Pronunciation (Incorrect)	GM Pronunciation (Correct)
coathanger	{coat}{hang}>er>	[kʌθˈəɪnʤə]	[koʊtˈhæŋə]
pothole	{pot}{hole}	[pɒˈθəl]	[pɒtˈhɒl]
goatherd	{goat}{herd}	[ˈgɒðəd]	[ˈgəʊtheɹd]
loophole	{loop}{hole}	[luˈfɒl]	[ˈluphəʊl]
upheld	{up}{held}	[ˈʌfɛld]	[ʌpˈhɛld]
cowherd	{cow}{herd}	[ˈkaʊɛɹd]	[kaʊˈhɛɹdˈ]
gigabytes	<giga<{byte}>s>	[gɪˈgaːbɪts]	[gɪgəˈbaɪts]
wobbliest	{wobble}>y>>est>	[ˈwɒblɪst]	[wɒˈbliɛst]
optimisers	{optim==ise}>er>>	[ˈoptɪmɪzəz]	[optɪˈmaɪzəz]
synchronizable	{syn==chron==ize}>able>	[ˌsɪˈʧɹaɪzəbɫ]	[ˌsiŋkreʊˈnaɪzəbɫ]

upheld

More samples: http://homepages.inf.ed.ac.uk/s1649890/morph/

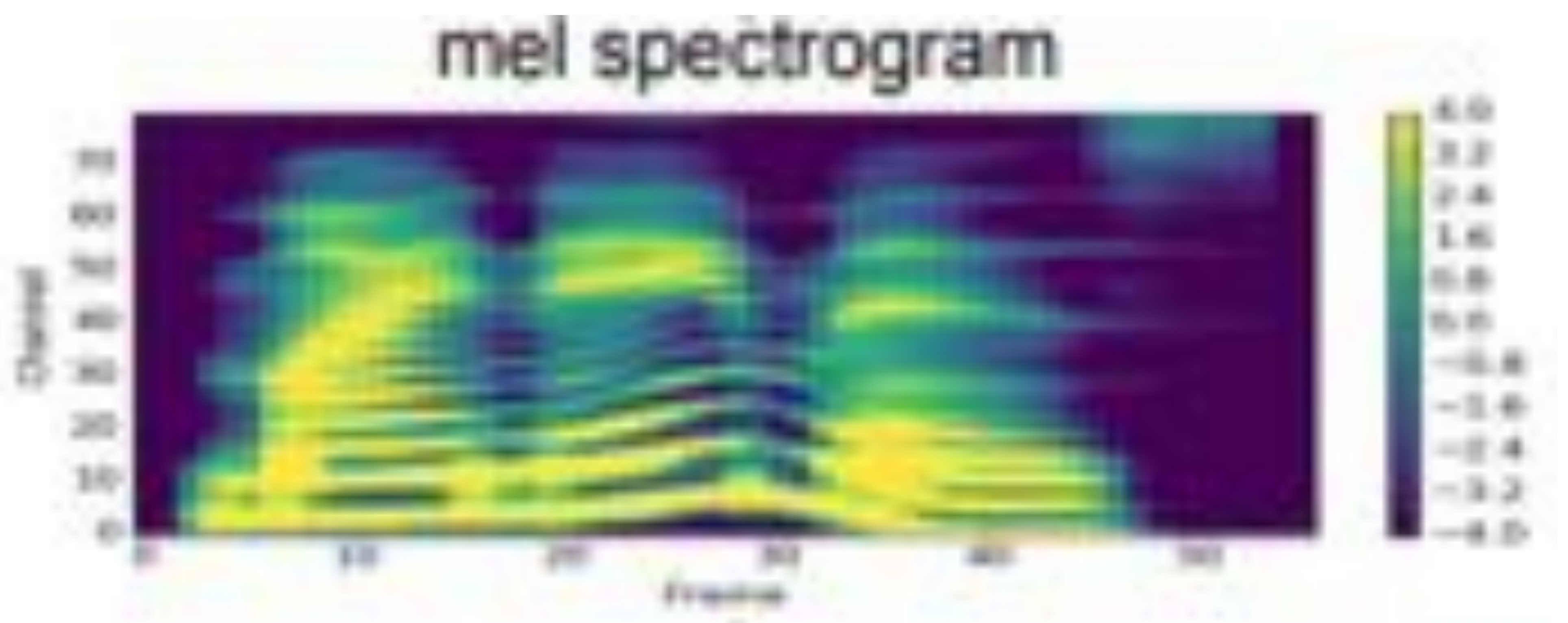
Outline

- Tutorial
  - Text processing
  - Regression
  - Waveform generation
- Current research
  - Waveform generation
  - Regression
  - Text processing
- <u>What next?</u>

Make use of rich linguistic structure

+ morphology
+ syntax
+ semantics
+ ... ?

Choose what to optimise at each stage



Regain controllability in waveform generation