Does ‘end-to-end’ speech synthesis make any sense?

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

Simon King

- Prof. of Speech Processing
- Director of the Centre for Speech Technology Research

- CSTR website: **`www.cstr.ed.ac.uk`**

- Teaching website: **`speech.zone`**

Motivation

text **speech**

http://proceedings.mlr.press/v70/arik17a.html (ICML 2017) — Deep Voice

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

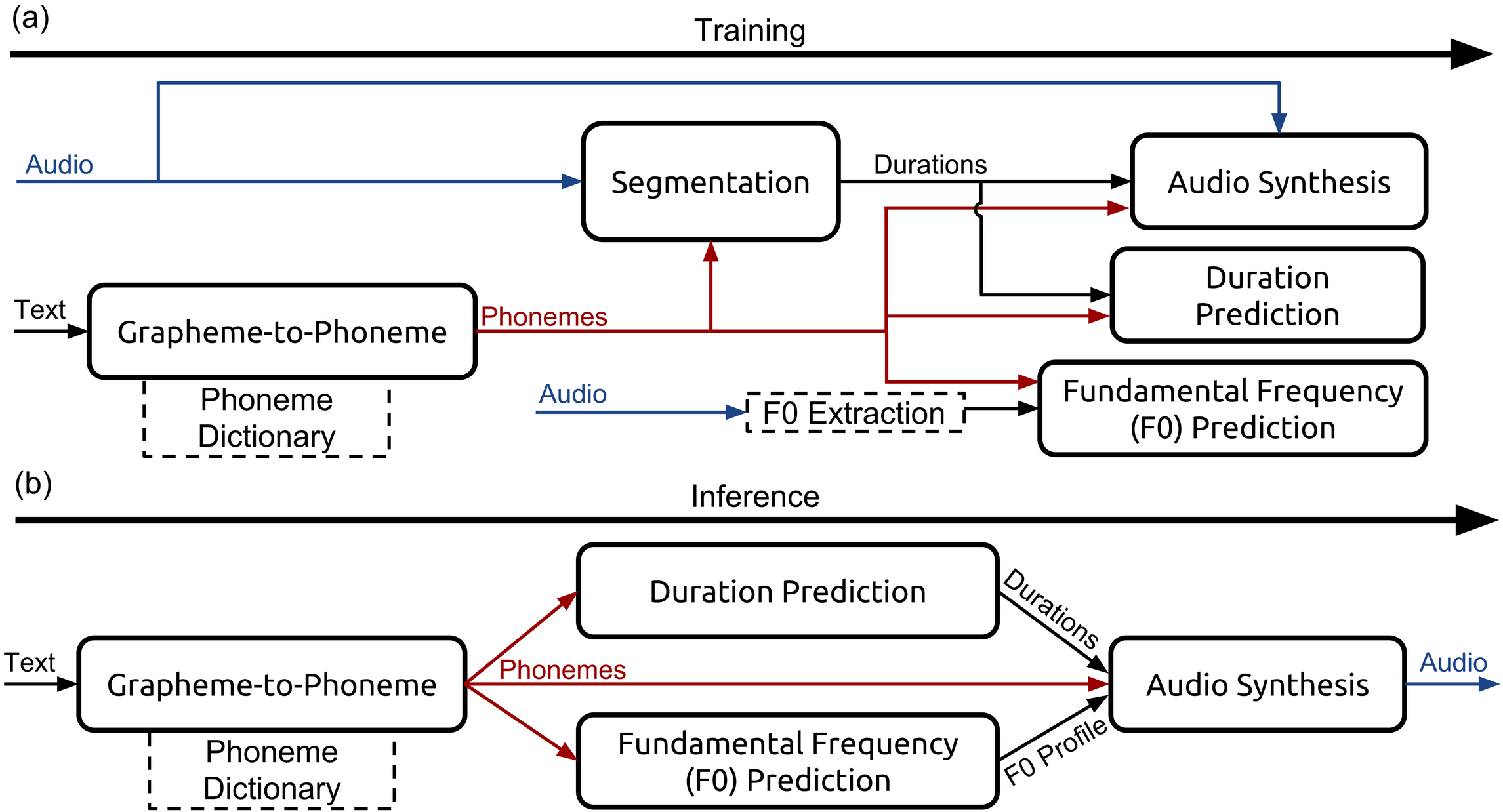
Sercan Ö. Arık [*1] **Mike Chrzanowski** [*1] **Adam Coates** [*1] **Gregory Diamos** [*1] **Andrew Gibiansky** [*1]
Yongguo Kang [*2] **Xian Li** [*2] **John Miller** [*1] **Andrew Ng** [*1] **Jonathan Raiman** [*1] **Shubho Sengupta** [*1]
Mohammad Shoeybi [*1]

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



DOI: 10.21437/Interspeech.2017-1452 — Tacotron

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

Tacotron: Towards End-to-End Speech Synthesis

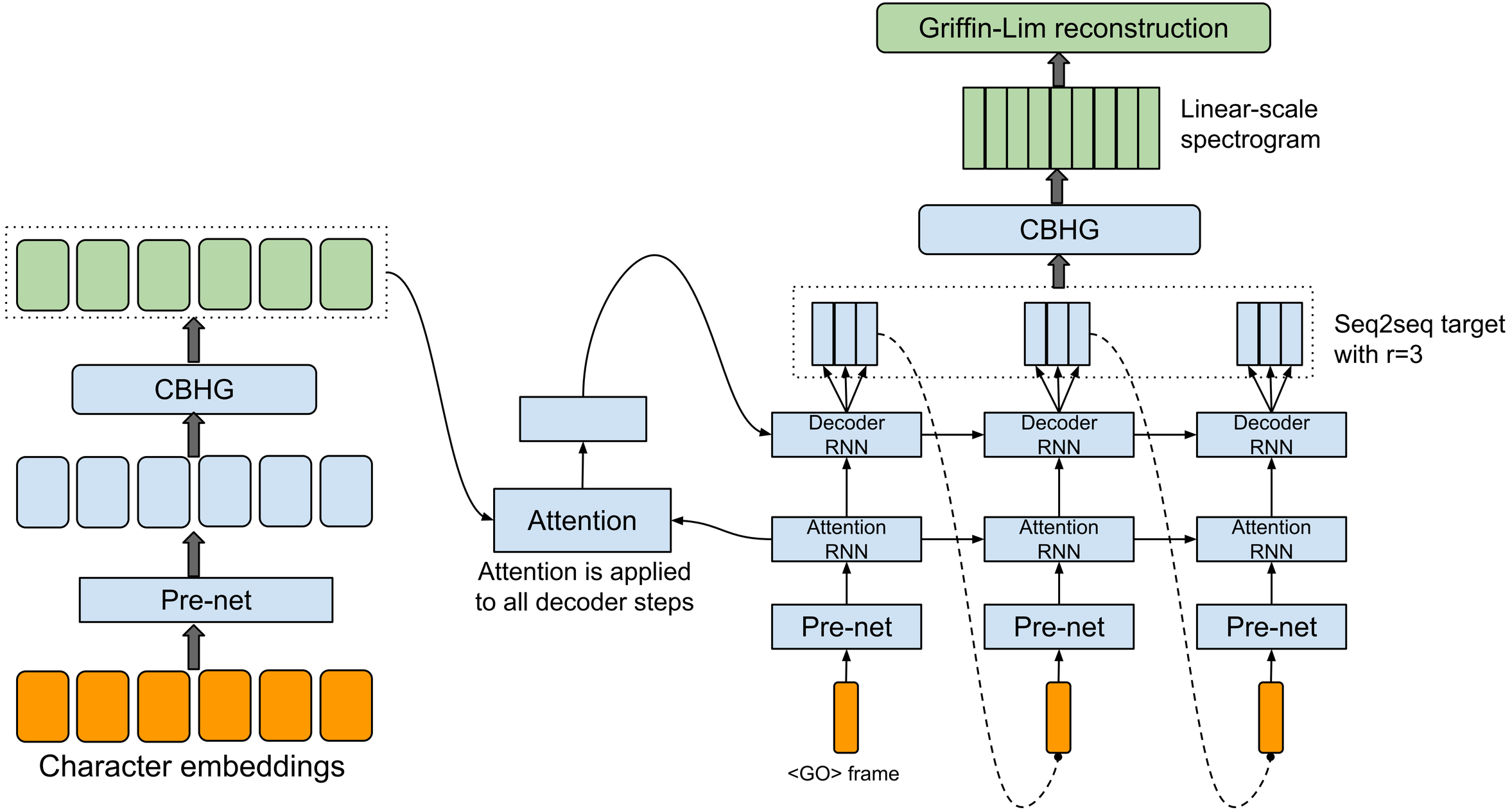
Yuxuan Wang, RJ Skerry-Ryan*, Daisy Stanton, Yonghui Wu, Ron J. Weiss†, Navdeep Jaitly, Zongheng Yang, Ying Xiao*, Zhifeng Chen, Samy Bengio†, Quoc Le, Yannis Agiomyrgiannakis, Rob Clark, Rif A. Saurous**

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-



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]

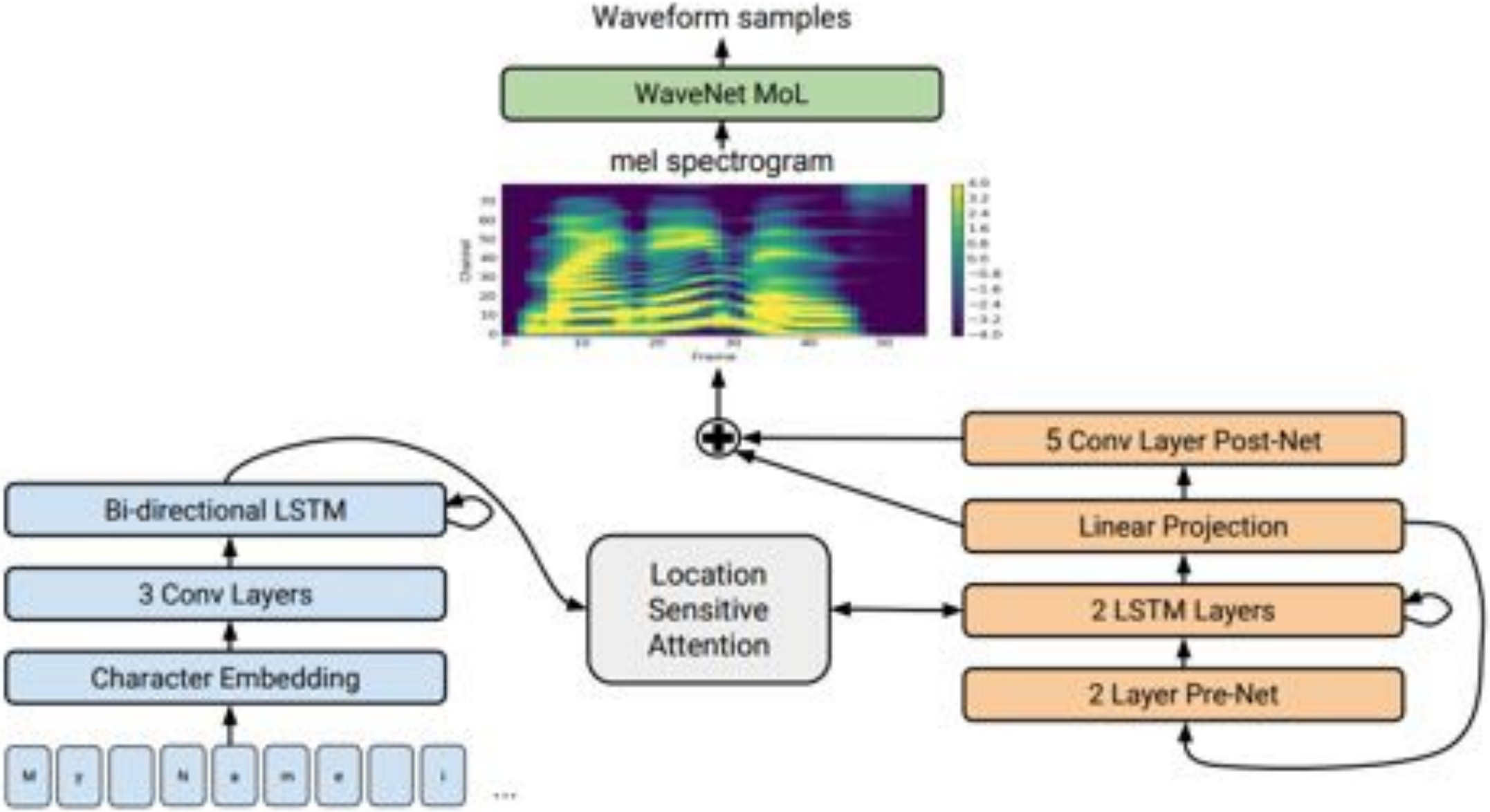
[1]Google, Inc., [2]University of California, Berkeley,
{jonathanasdf,rpang,yonghui}@google.com

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



WAVENET: A GENERATIVE MODEL FOR RAW AUDIO

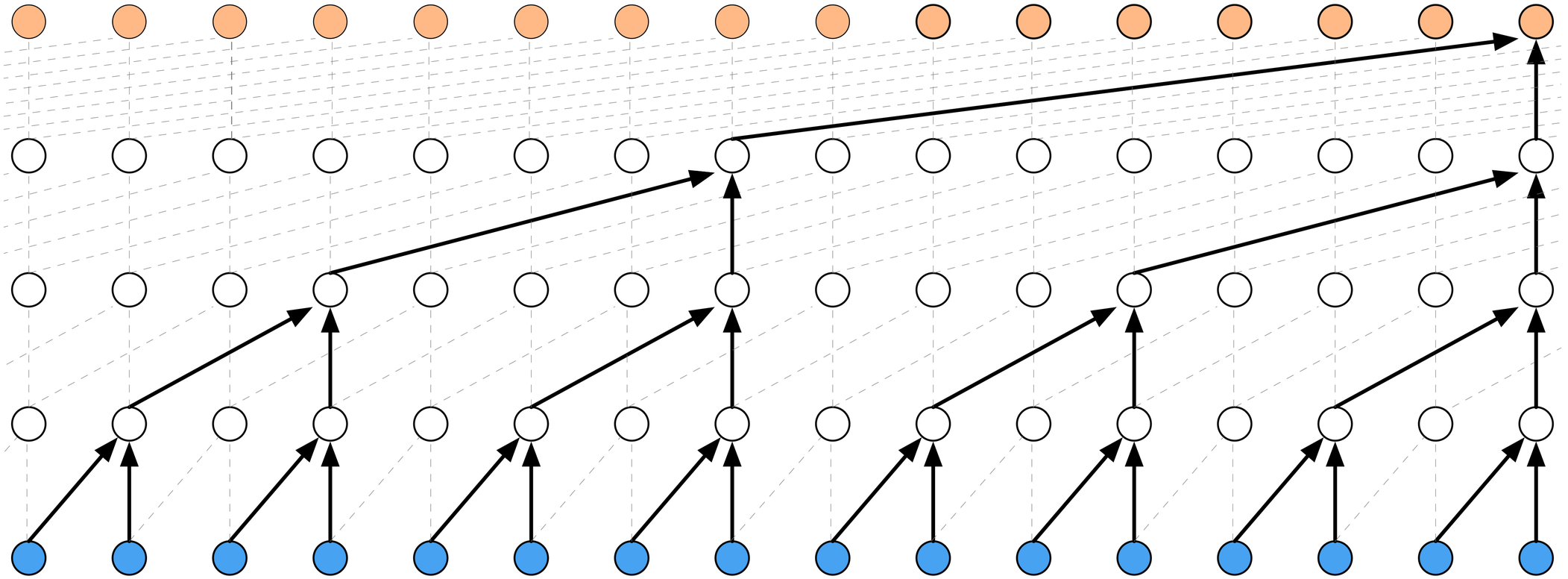
Aäron van den Oord **Sander Dieleman** **Heiga Zen**[†]

Karen Simonyan **Oriol Vinyals** **Alex Graves**

Nal Kalchbrenner **Andrew Senior** **Koray Kavukcuoglu**

{avdnoord, sedielem, heigazen, simonyan, vinyals, gravesa, nalk, andrewsenior, korayk}@google.com
Google DeepMind, London, UK
[†] Google, London, UK

ABSTRACT



arXiv:1710.07654v3 — Deep Voice 3

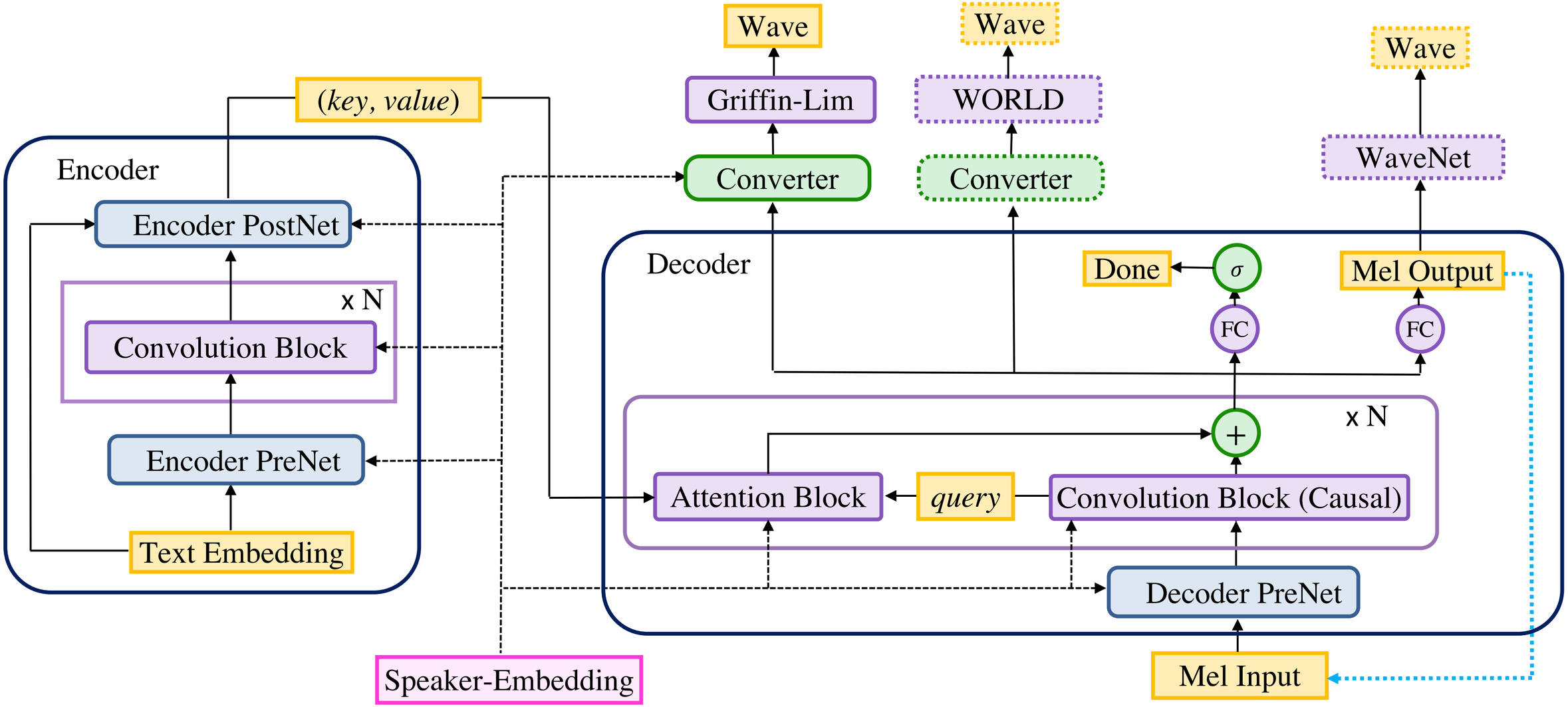
Published as a conference paper at ICLR 2018

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

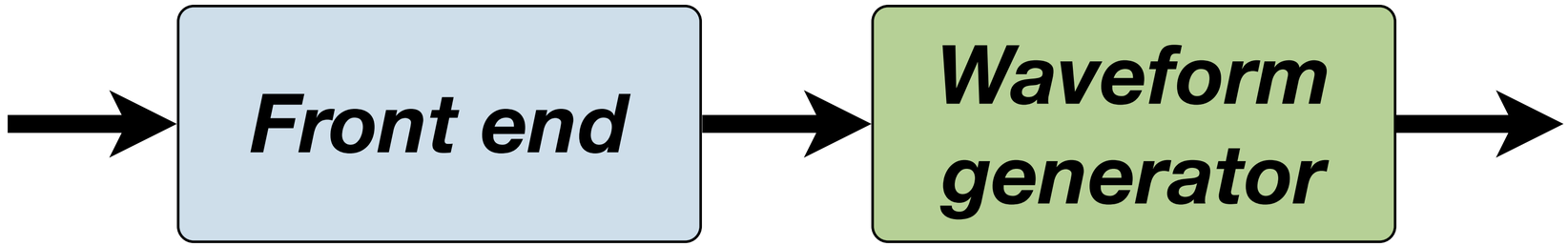


Contents

1. "Traditional" methods
2. Here comes machine learning
3. The best of both

Part 1 — "Traditional" methods — the text-to-speech pipeline

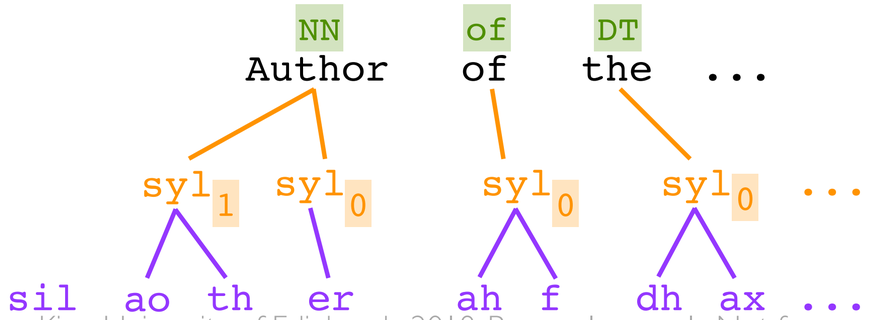
The classic two-stage pipeline of unit selection



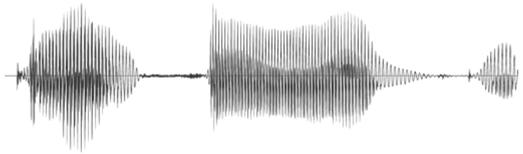
text

`Author of the…`

linguistic specification



waveform

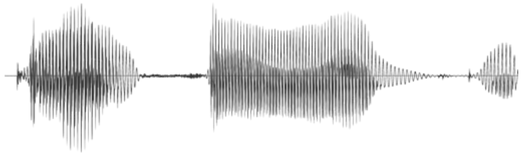


The end-to-end problem we want to solve

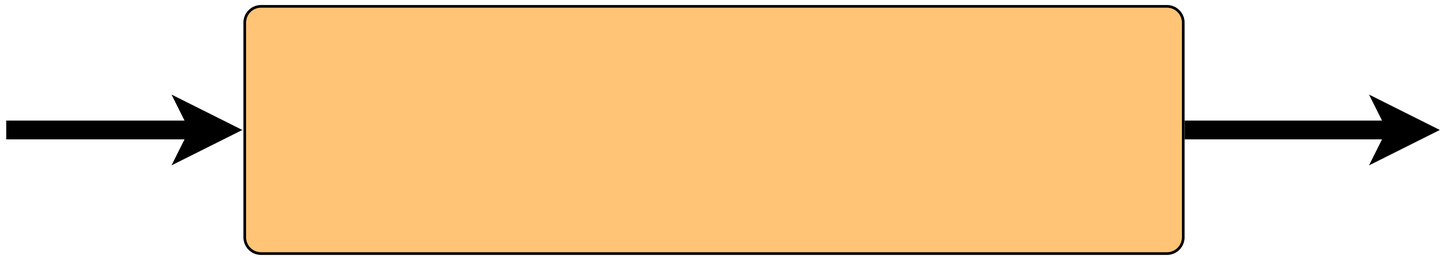
text

```
Author of the…
```

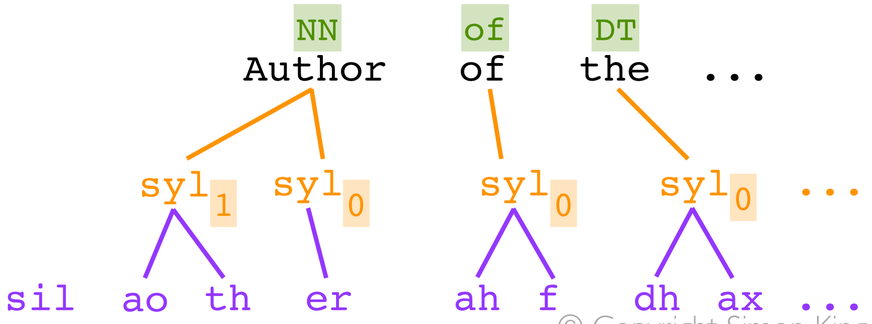
waveform



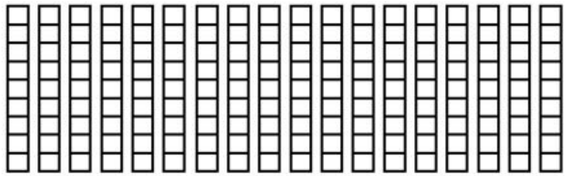
A problem we can actually solve with machine learning



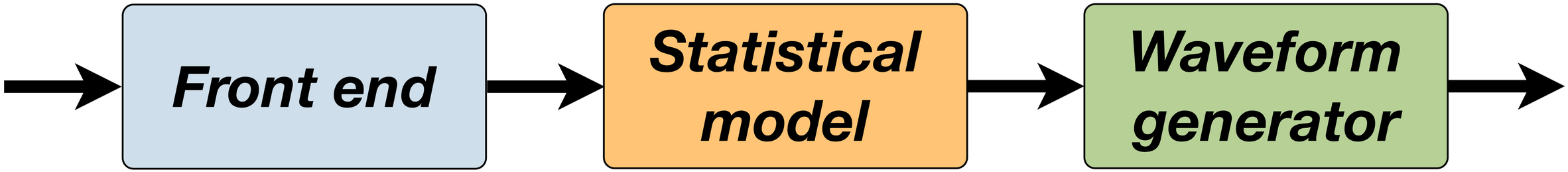
linguistic specification



acoustic features



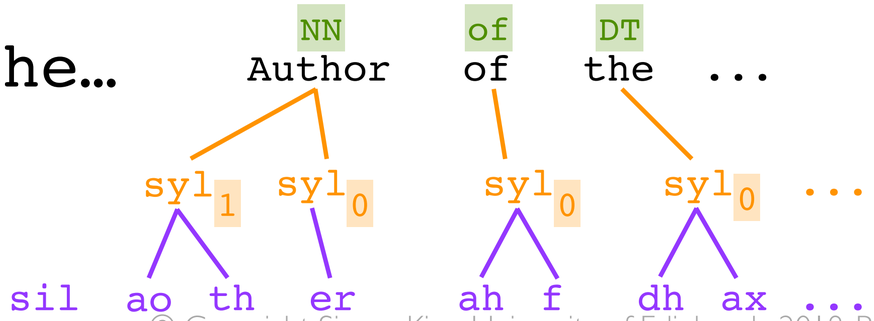
The classic three-stage pipeline of statistical parametric speech synthesis



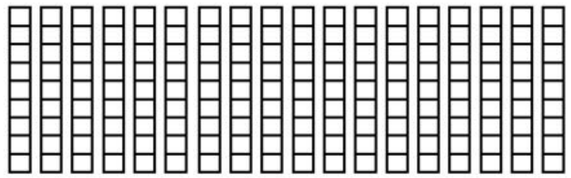
text

Author of the…

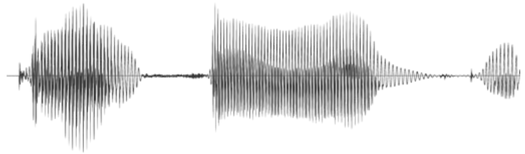
linguistic specification



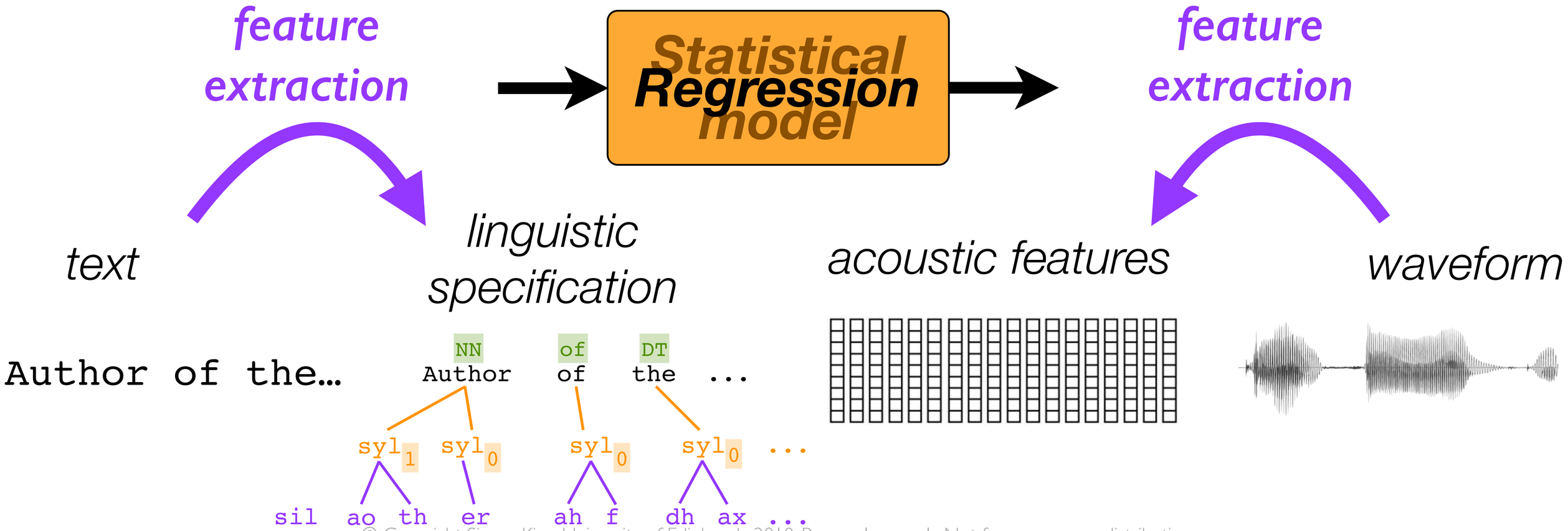
acoustic features



waveform

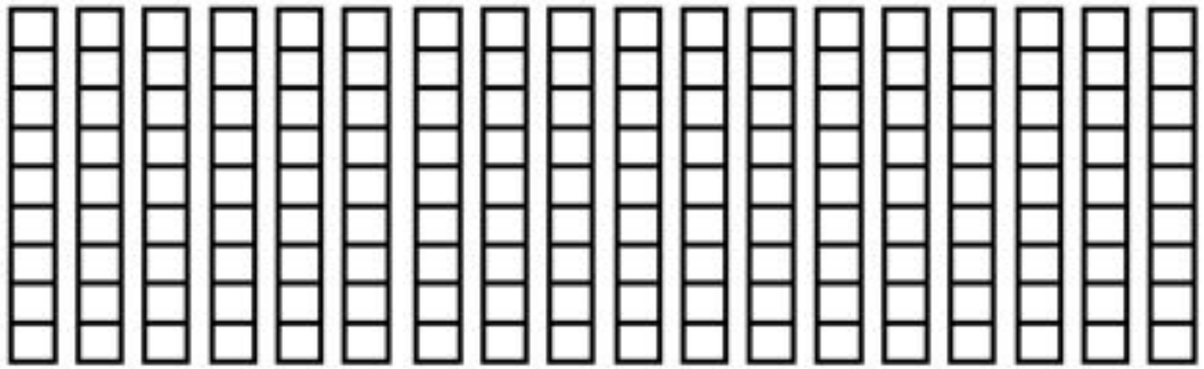


The classic three-stage pipeline of statistical parametric speech synthesis



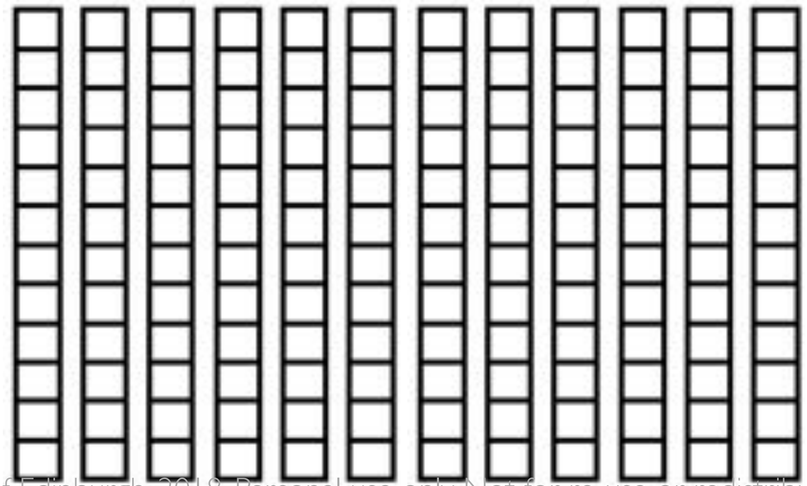
We can describe the core problem as **sequence-to-sequence regression**

output sequence
(acoustic features)



Different lengths, because of differing ‘clock rates’

input sequence
(linguistic features)



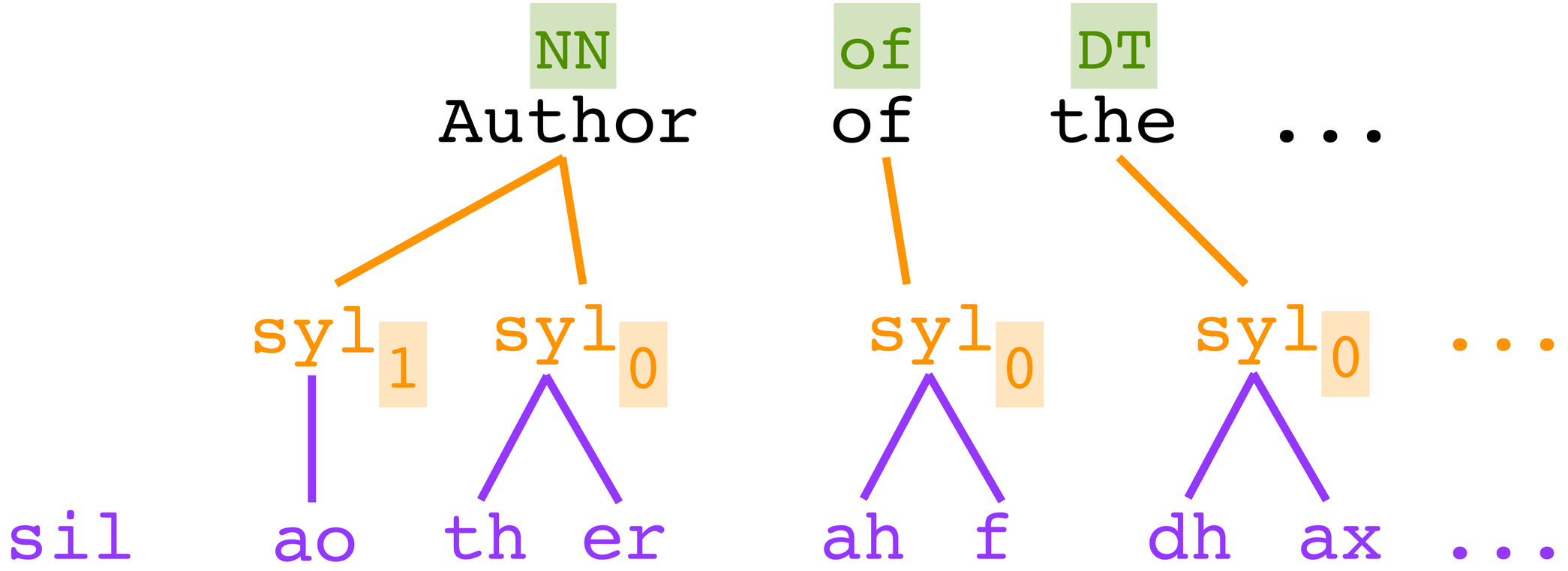
From text to speech

- Text processing
  - pipeline architecture
  - linguistic specification
- Regression
  - duration model
  - acoustic model
- Waveform generation
  - acoustic features
  - signal processing

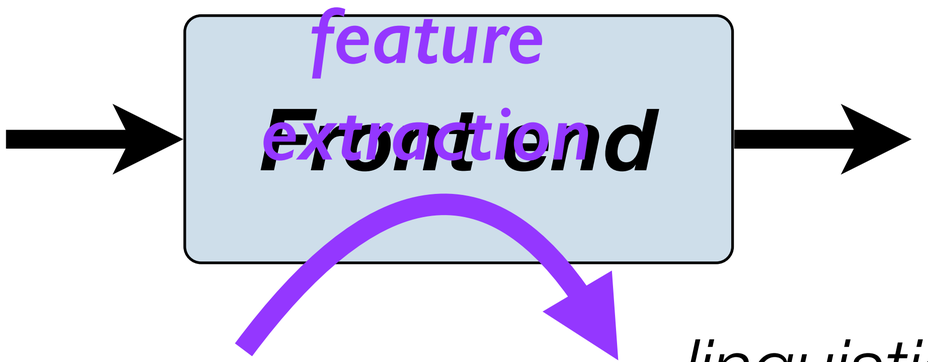
From text to speech

- Text processing
  - pipeline architecture
  - linguistic specification
- Regression
  - duration model
  - acoustic model
- Waveform generation
  - acoustic features
  - signal processing

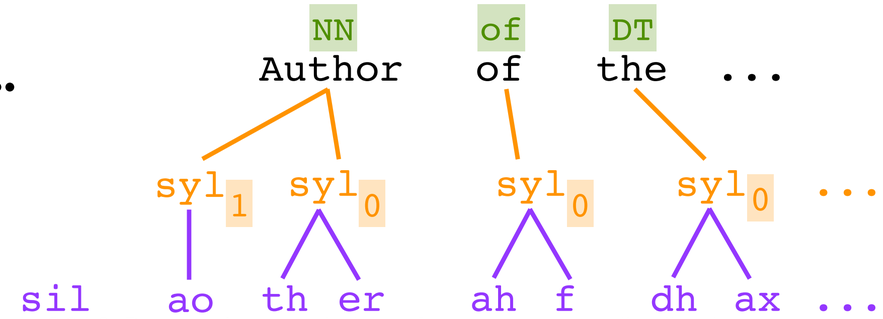
The linguistic specification



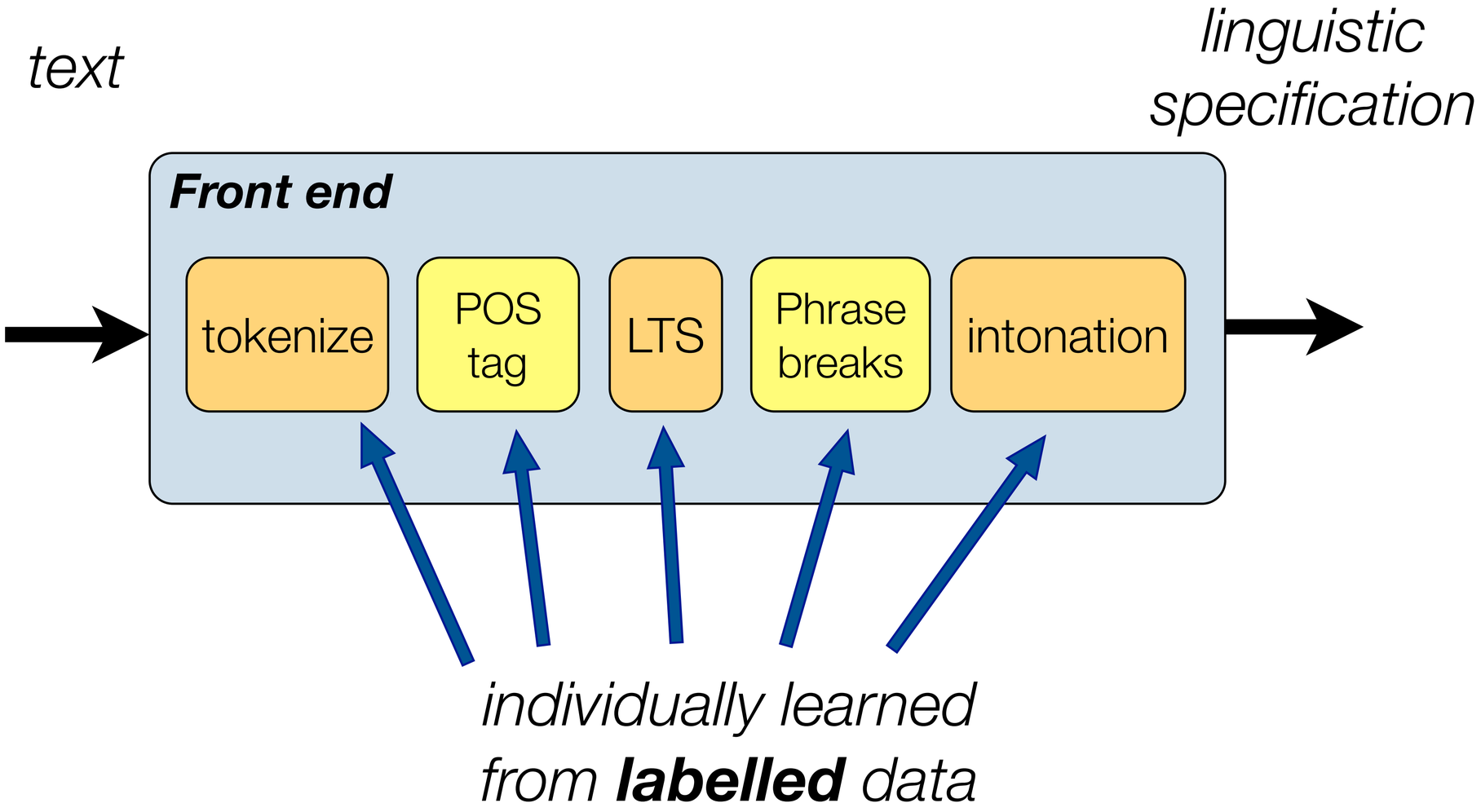
Extracting features from text using the front end



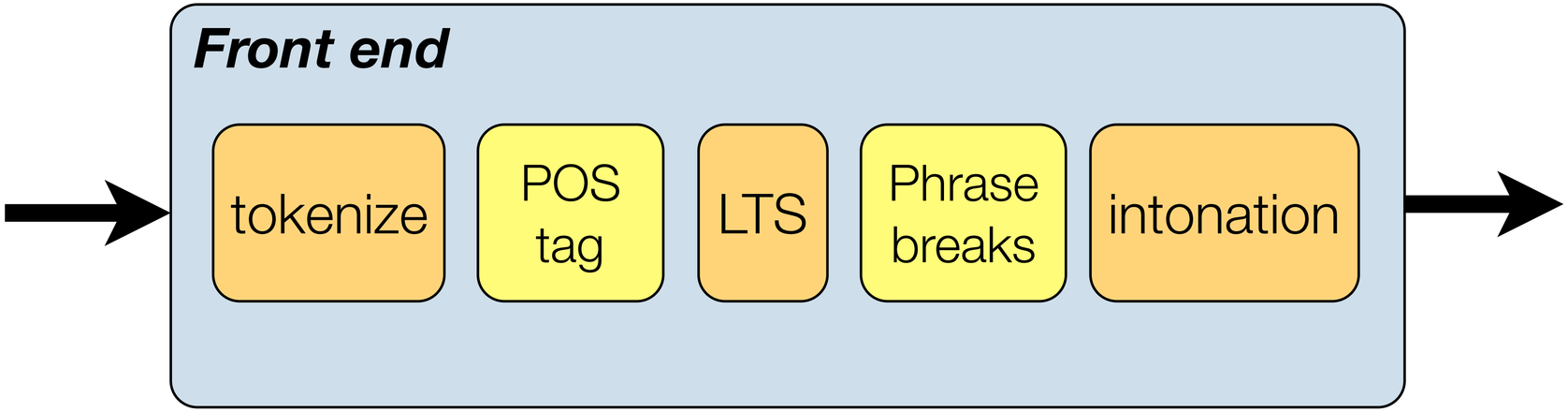
Author of the…



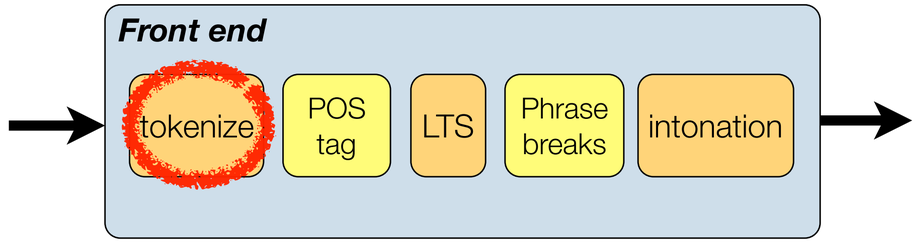
Text processing pipeline



Text processing pipeline



Tokenize & Normalize

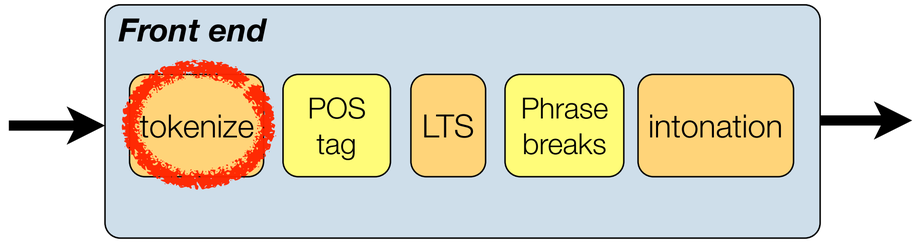


- Step 1: divide input stream into tokens, which are potential words

- 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

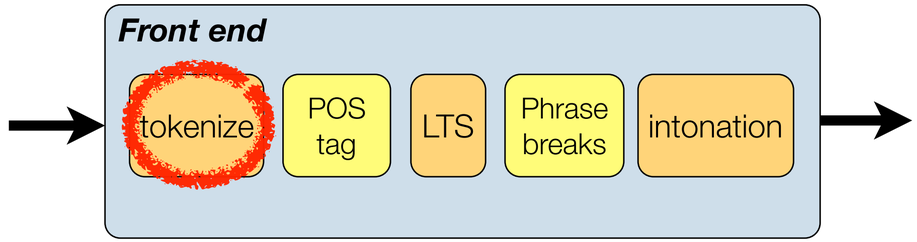


- Step 2: classify every token, finding **Non-Standard Words** that need further processing

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

   NYER                MONEY   ASWD    NUM LSEQ
```

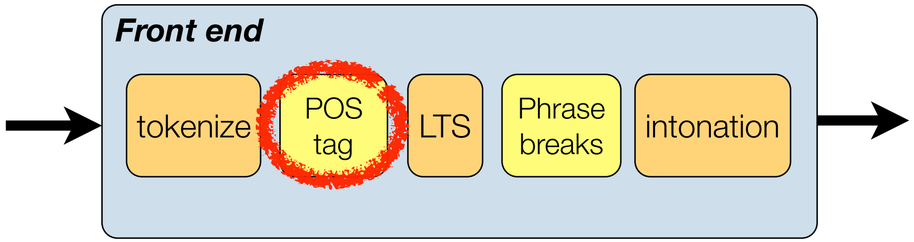
Tokenize & Normalize



- 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
```

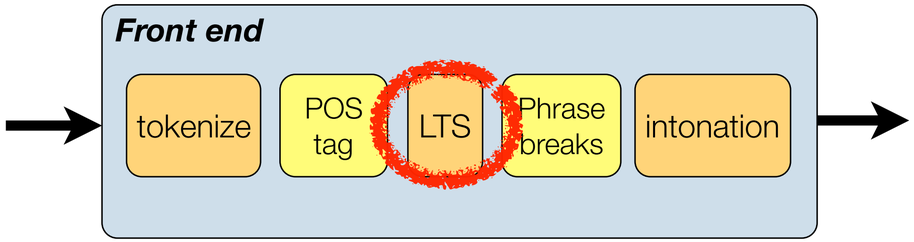
POS tagging



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

```
NN  Director
IN  of
DT  the
NP  McCormick
NP  Public
NPS Affairs
NP  Institute
IN  at
NP  U-Mass
NP  Boston,
NP  Doctor
NP  Ed
NP  Beard,
VBZ says
DT  the
NN  push
IN  for
VBP do
PP  it
PP  yourself
NN  lawmaking
```

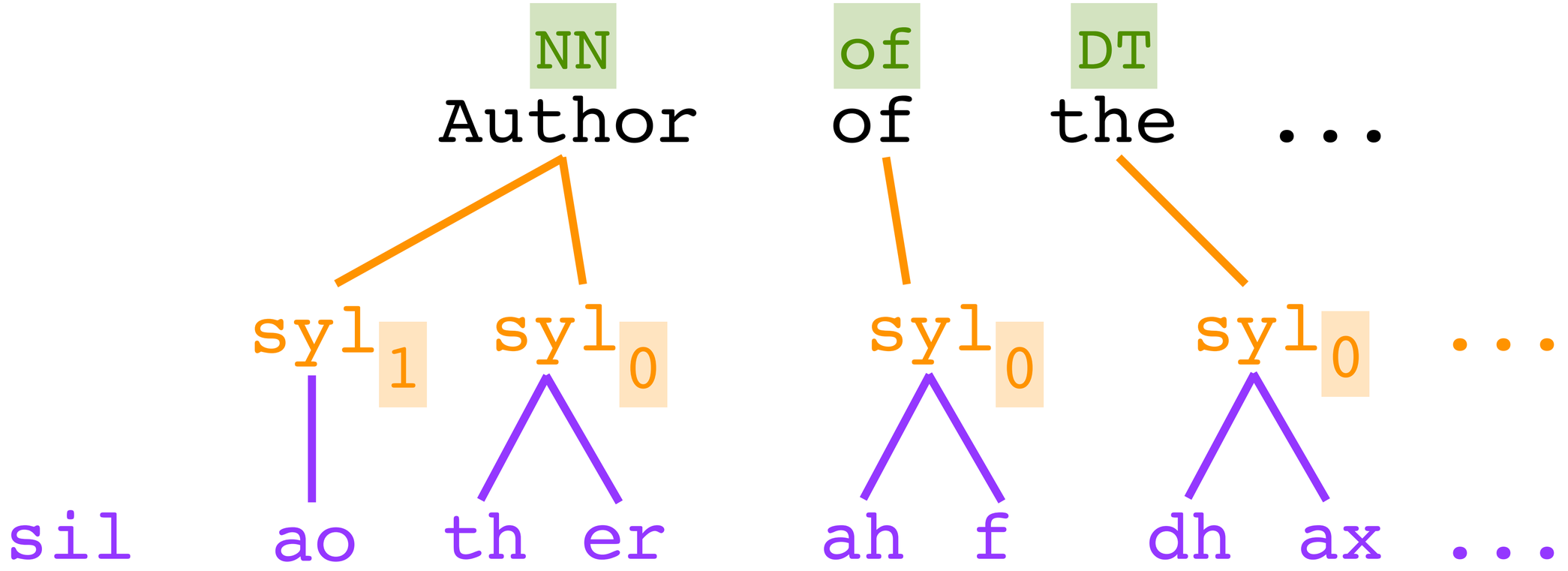
Pronunciation / LTS



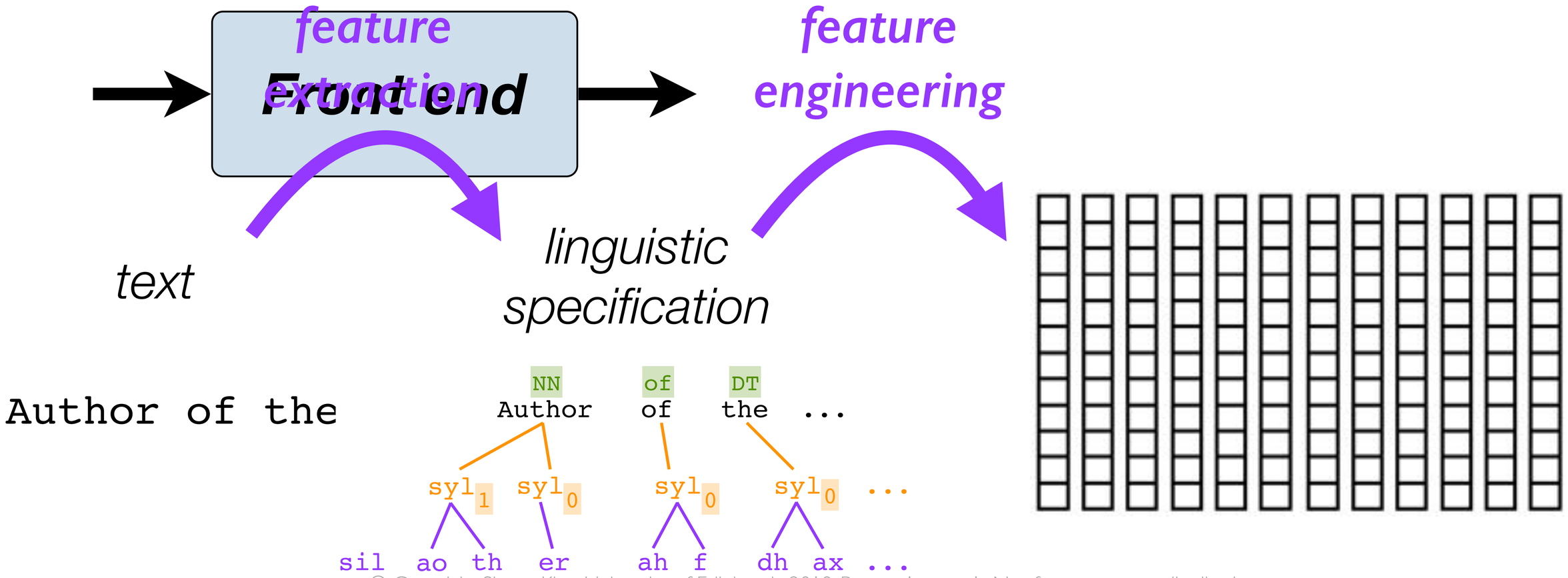
- Pronunciation model
  - dictionary look-up, *plus*
  - letter-to-sound model
- But
  - need deep **knowledge** of the language to design the phoneme set
  - human **expert** must write dictionary

```
ADVOCATING  AE1 D V AH0 K EY2 T IH0 NG
ADVOCATION  AE2 D V AH0 K EY1 SH AH0 N
ADWEEK  AE1 D W IY0 K
ADWELL  AH0 D W EH1 L
ADY  EY1 D IY0
ADZ  AE1 D Z
AE  EY1
AEGEAN  IH0 JH IY1 AH0 N
AEGIS  IY1 JH AH0 S
AEGON  EY1 G AA0 N
AELTUS  AE1 L T AH0 S
AENEAS  AE1 N IY0 AH0 S
AENEID  AH0 N IY1 IH0 D
AEQUITRON  EY1 K W IH0 T R AA0 N
AER  EH1 R
AERIAL  EH1 R IY0 AH0 L
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
```

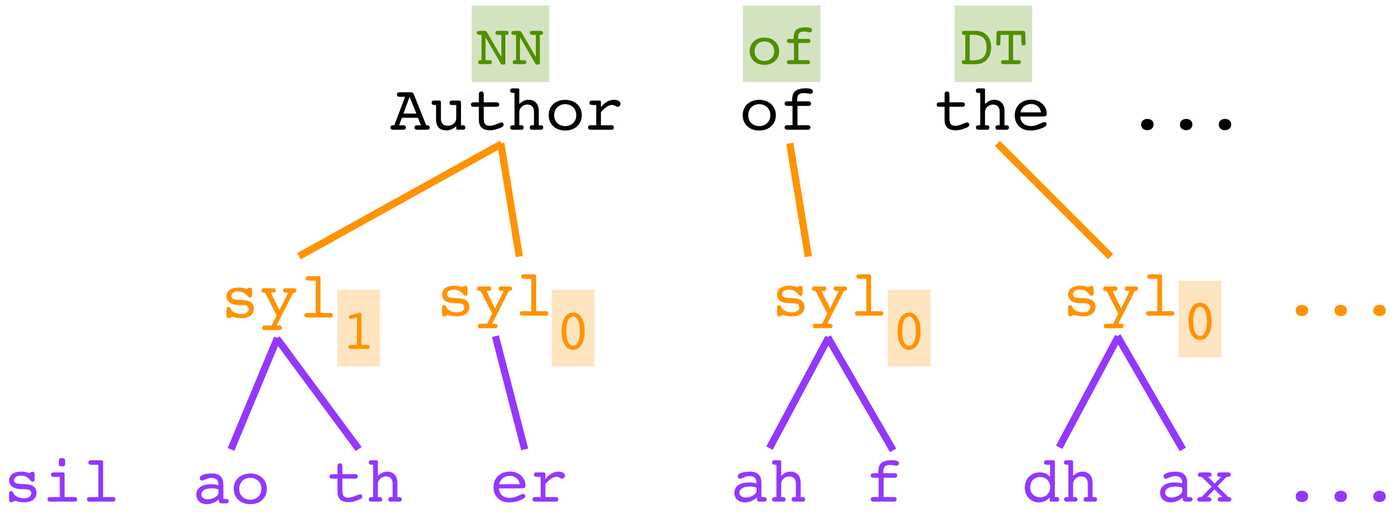
The linguistic specification

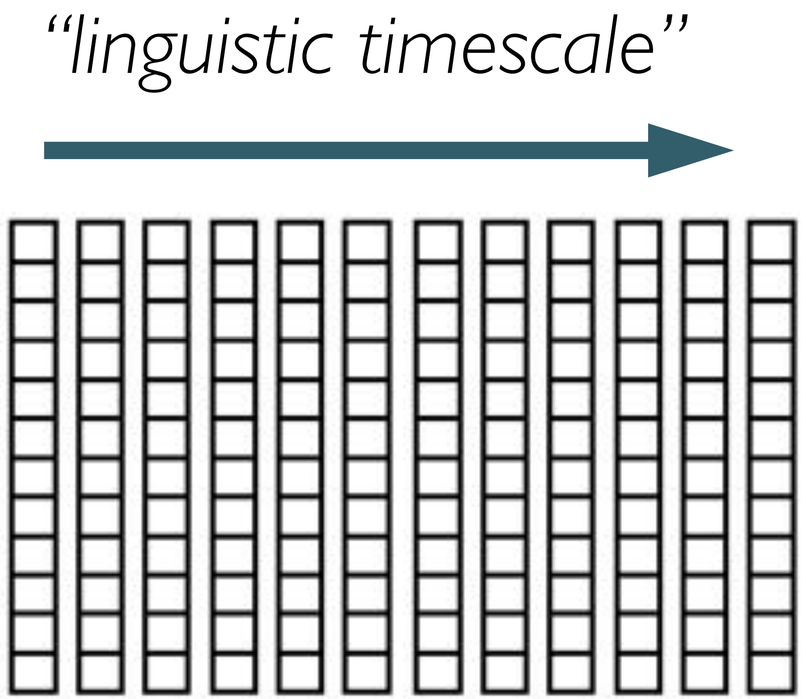


Linguistic feature engineering

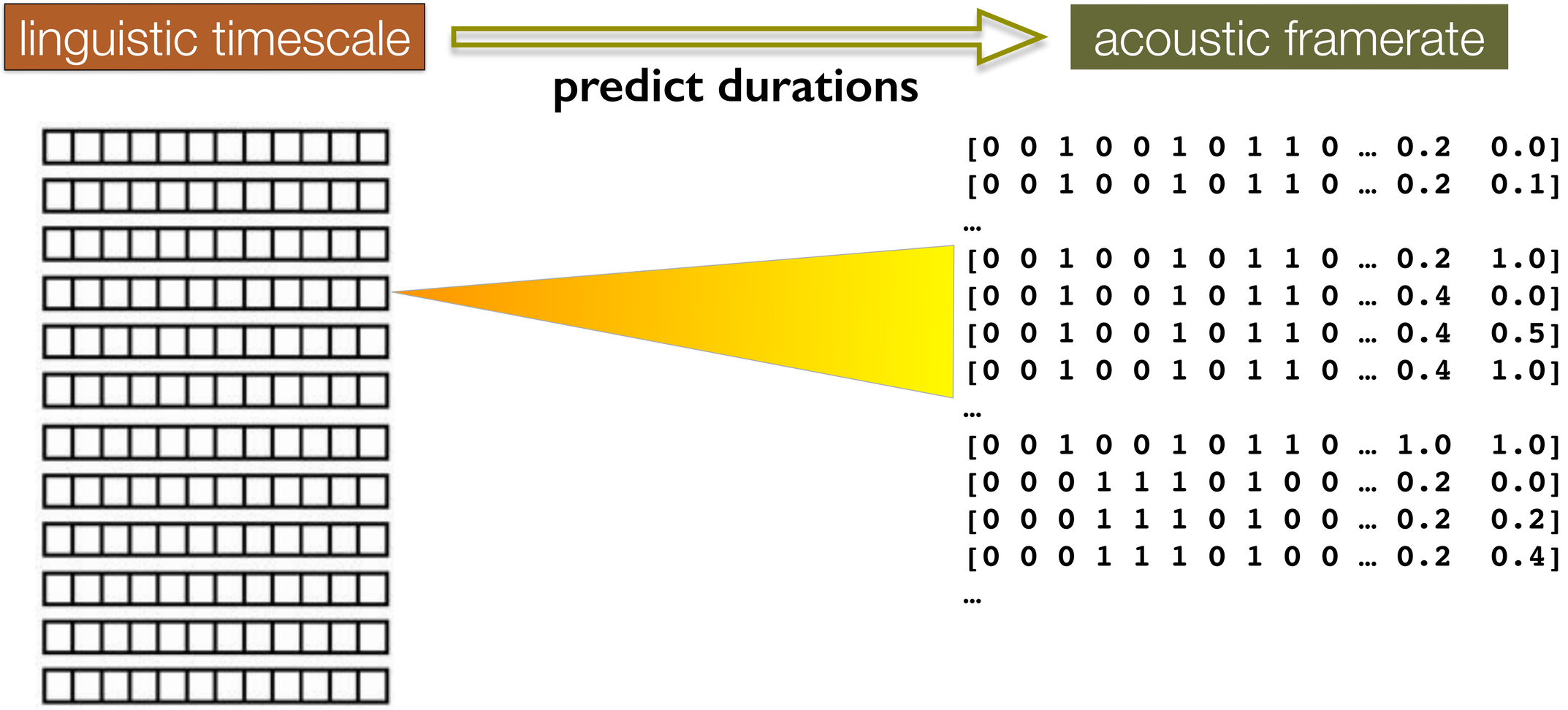


Flatten & encode: convert linguistic specification to vector sequence



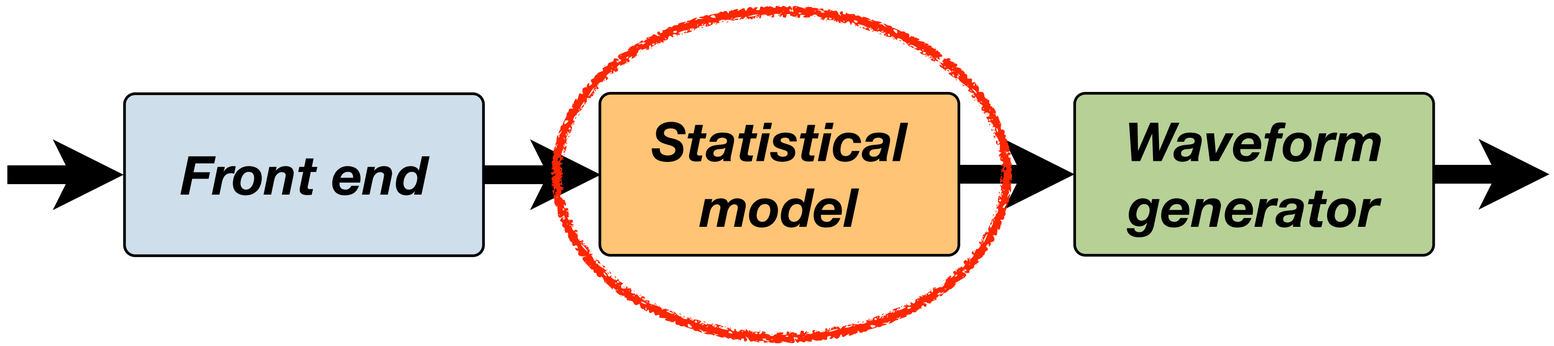


Upsample: add duration information

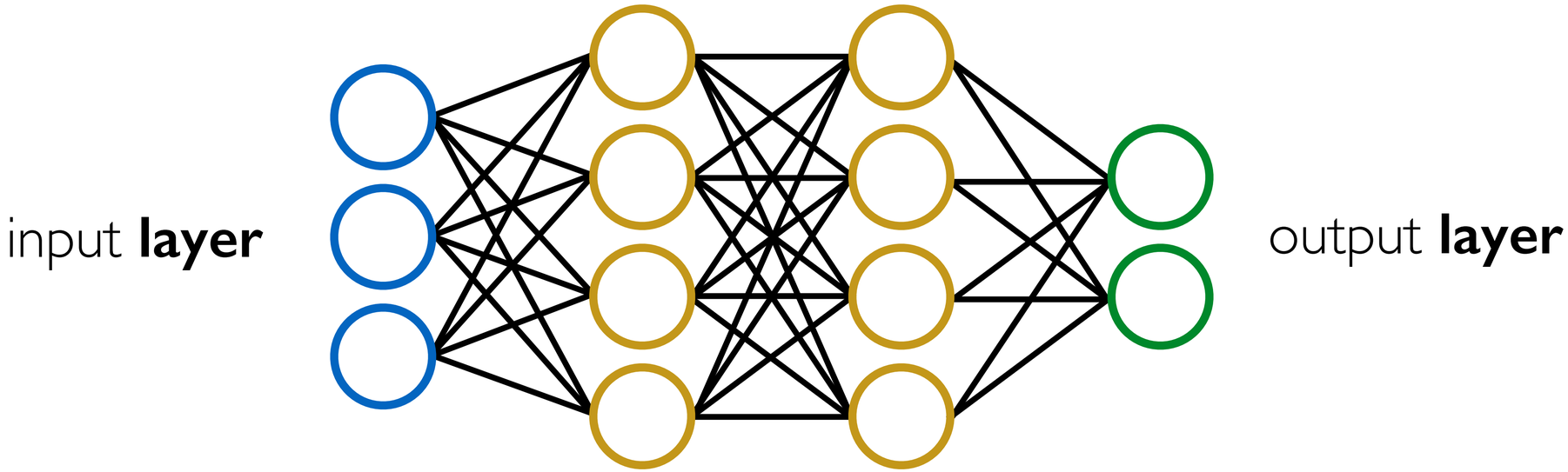


From text to speech

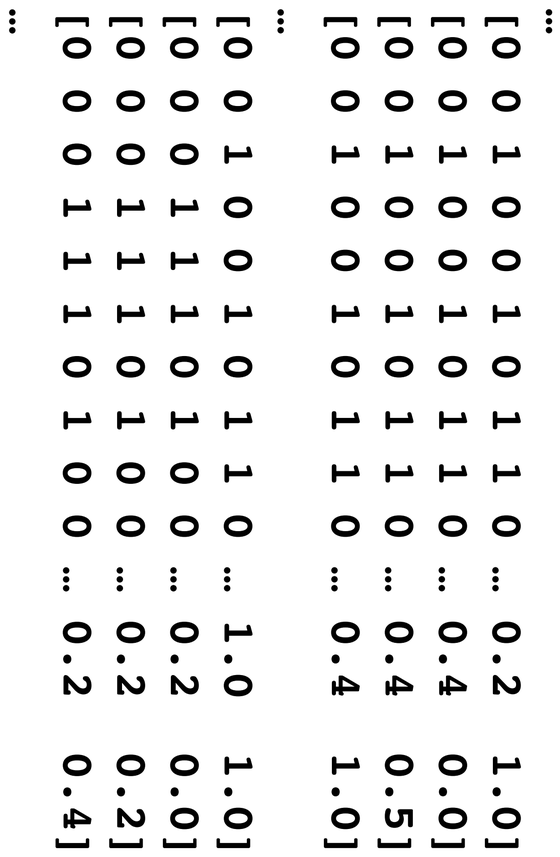
- Text processing
  - pipeline architecture
  - linguistic specification
- Regression
  - duration model
  - acoustic model
- Waveform generation
  - acoustic features
  - signal processing

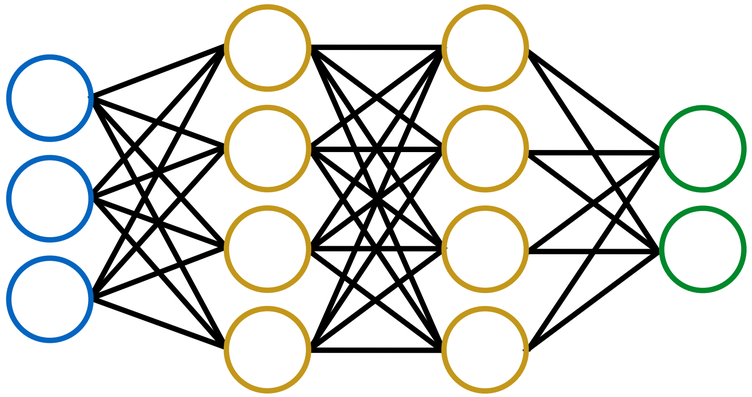


Acoustic model: a simple feed-forward neural network



Synthesis with a simple neural network — frame-by-frame

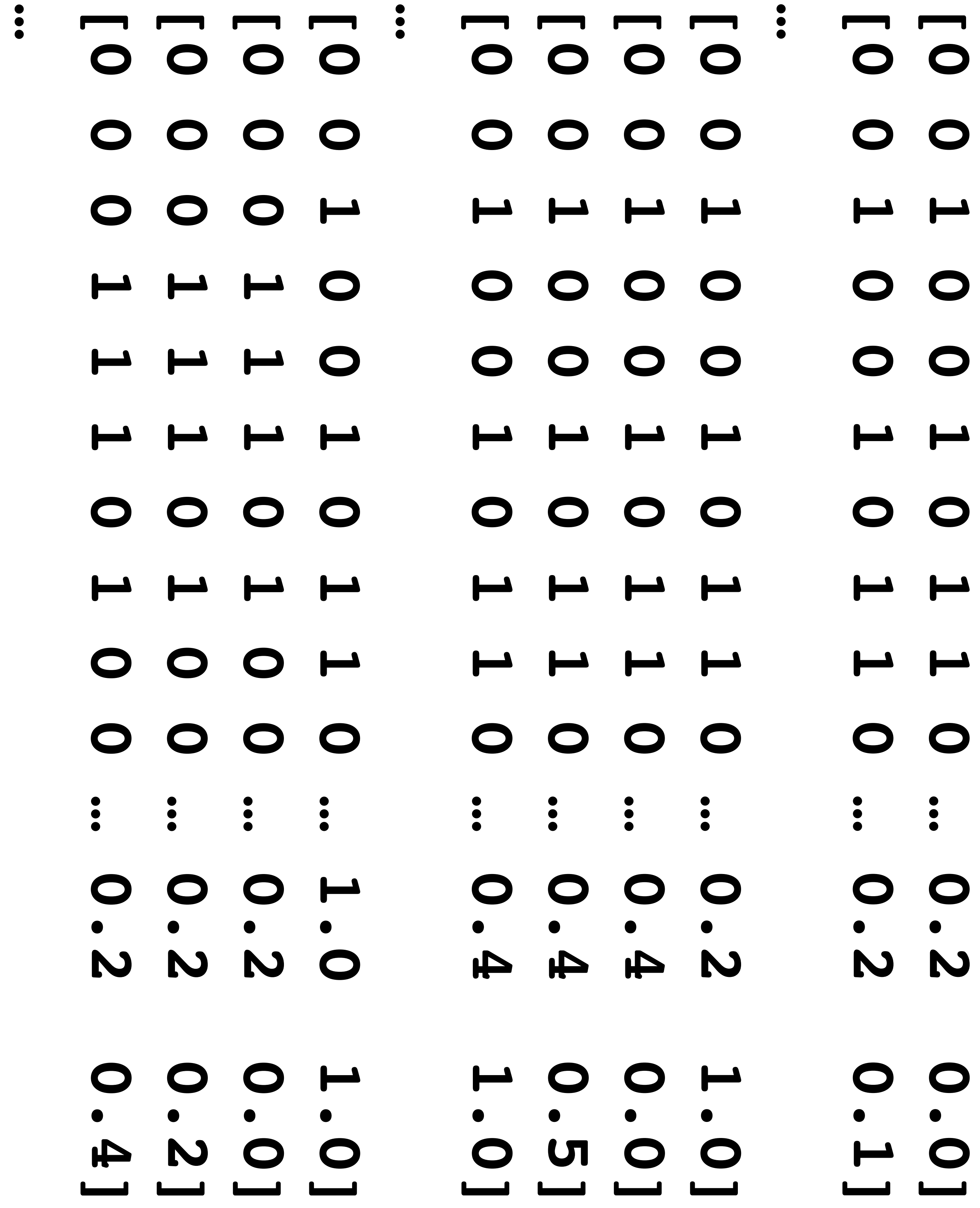


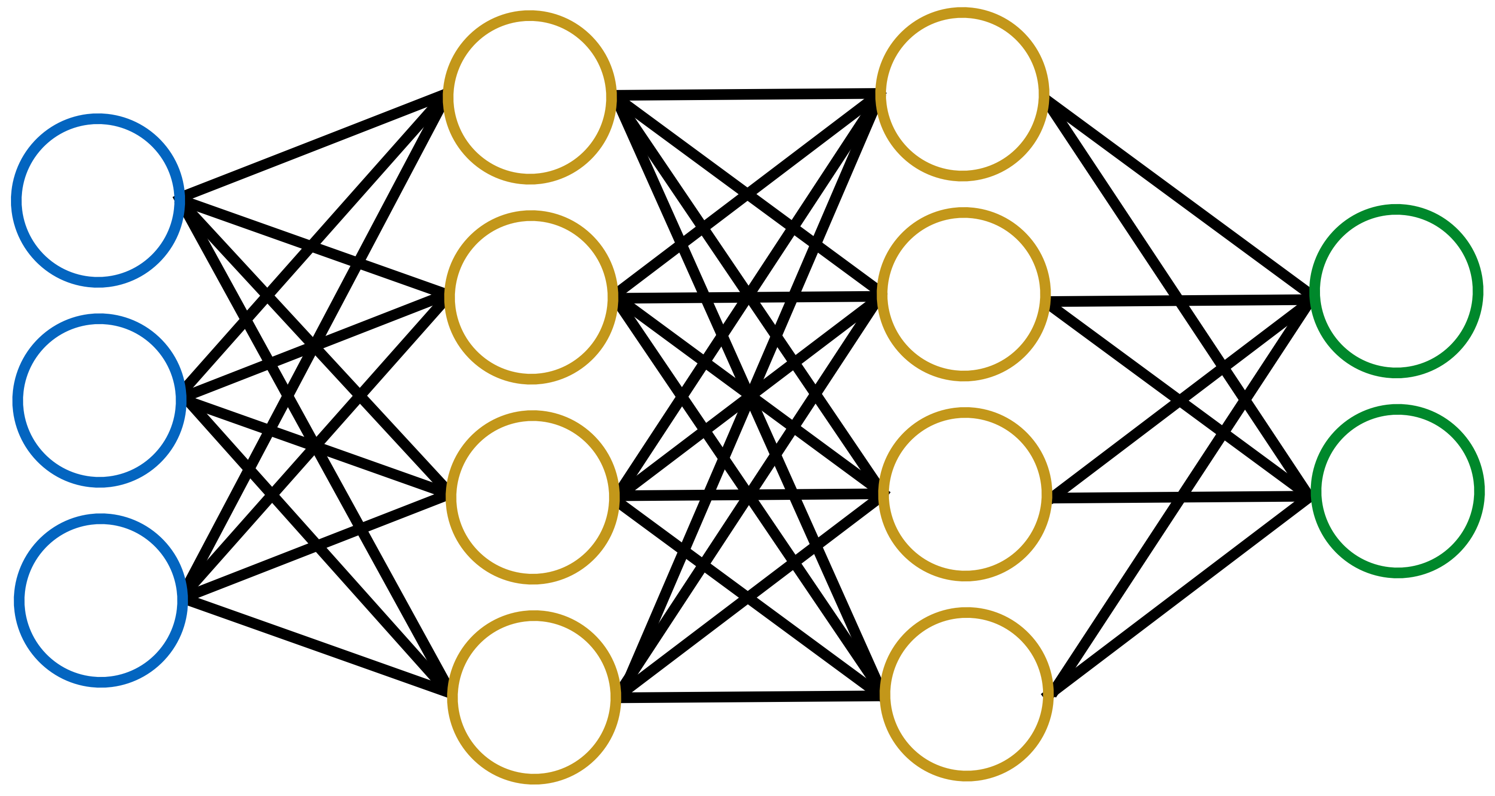


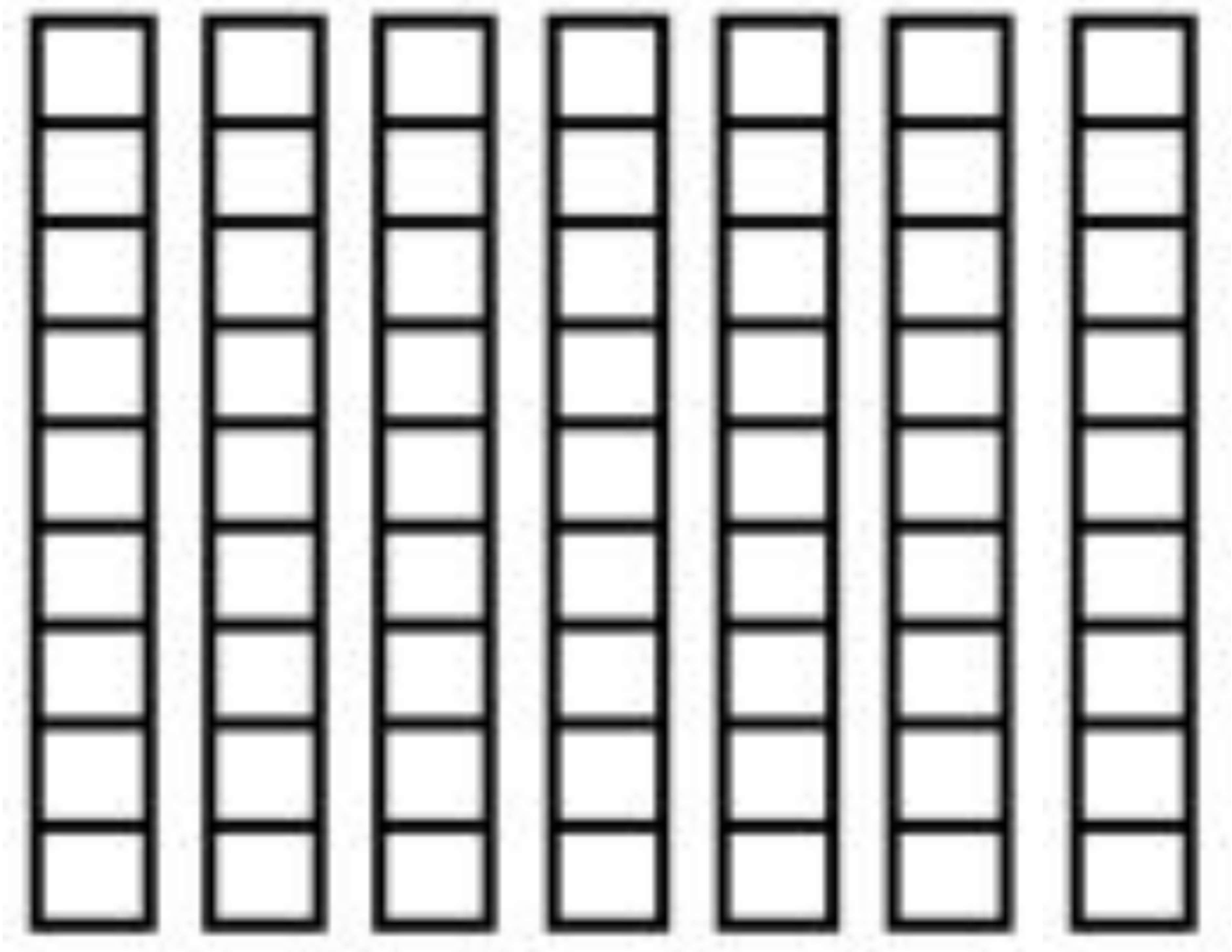
From text to speech

- Text processing
  - pipeline architecture
  - linguistic specification
- Regression
  - duration model
  - acoustic model
- Waveform generation
  - acoustic features
  - signal processing

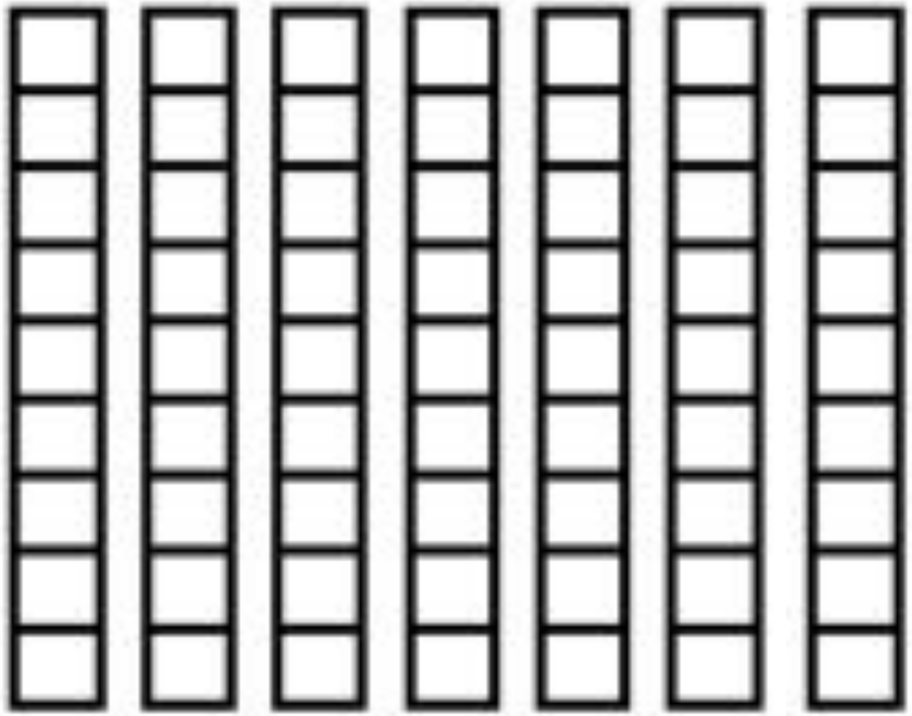
What are the acoustic features?

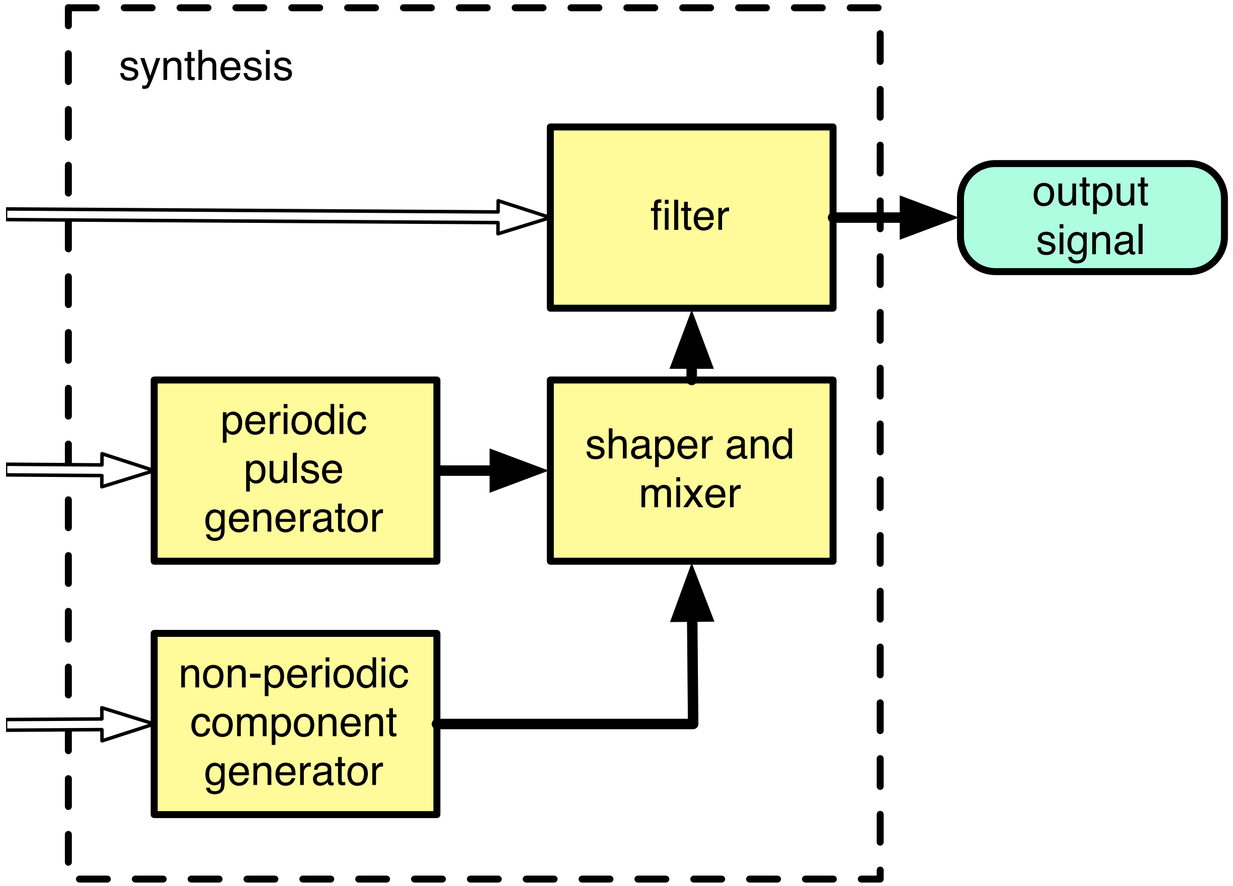




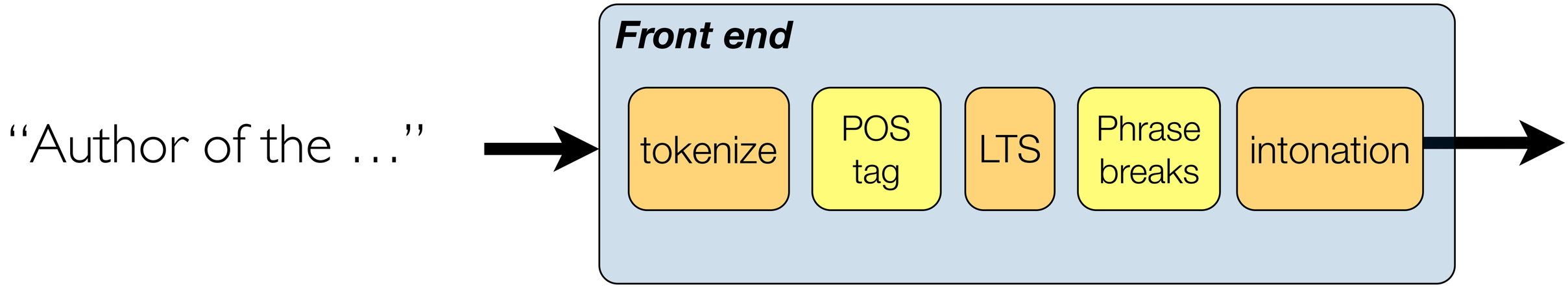


What are the acoustic features?

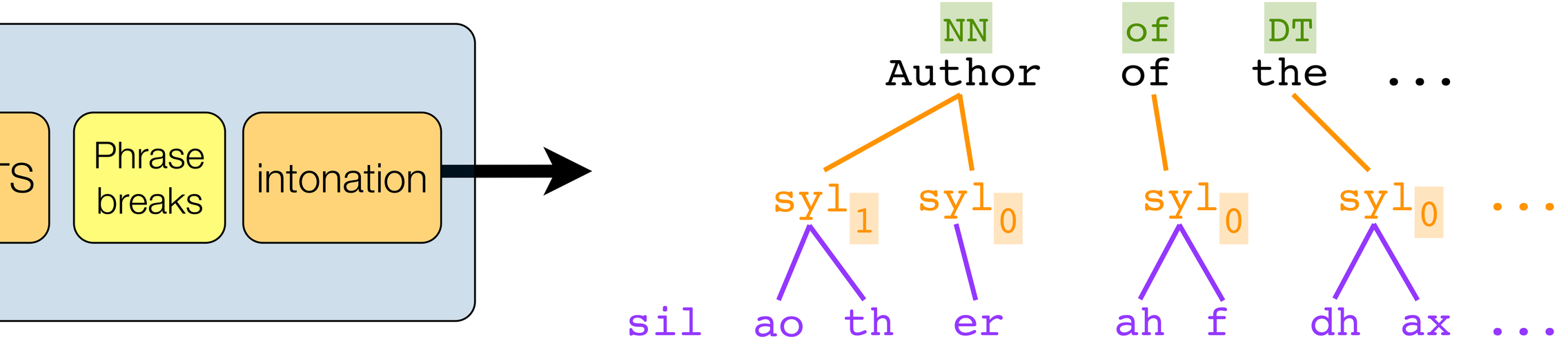




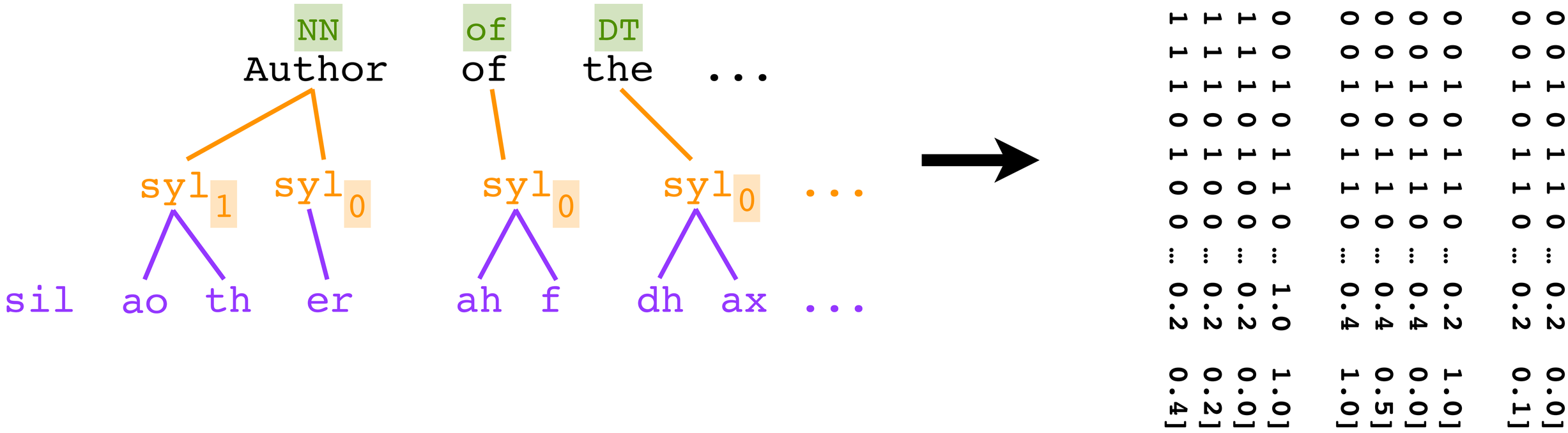
Putting it all together: text-to-speech with a neural network



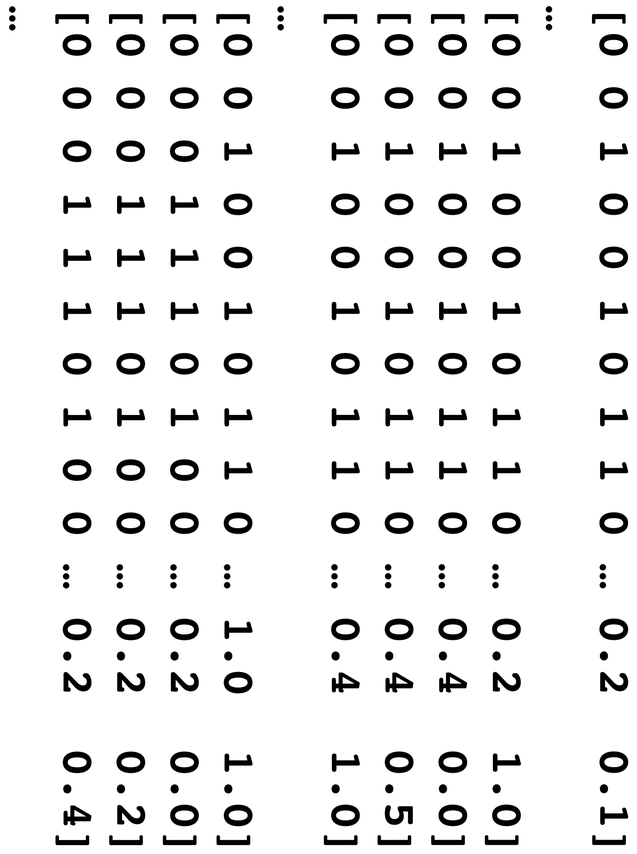
Putting it all together: text-to-speech with a neural network

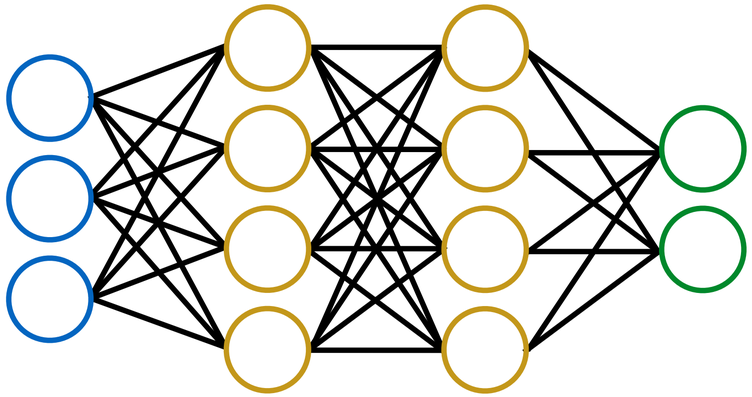


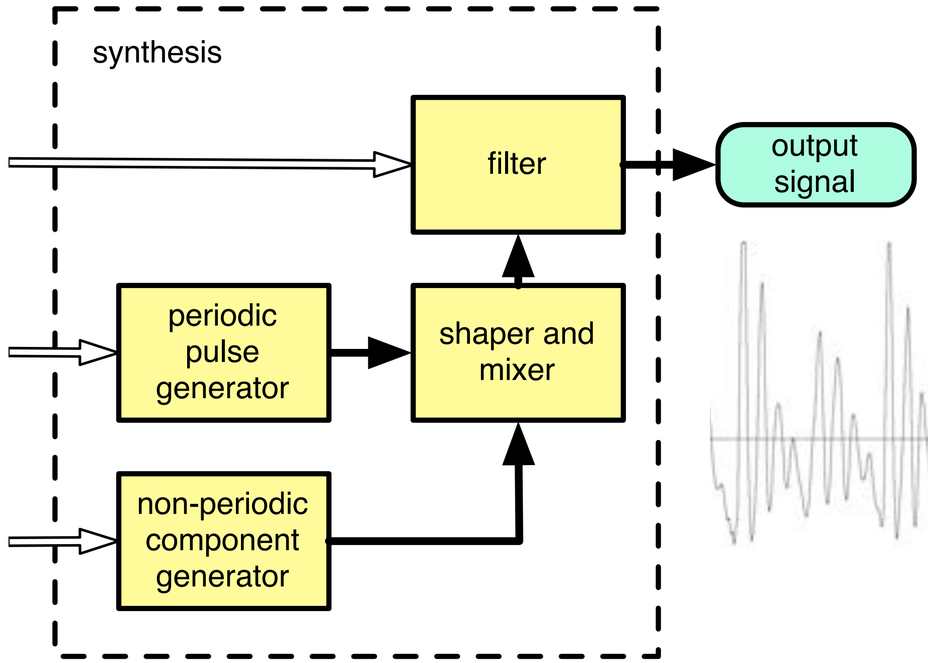
Putting it all together: text-to-speech with a neural network



Putting it all together: text-to-speech with a neural network







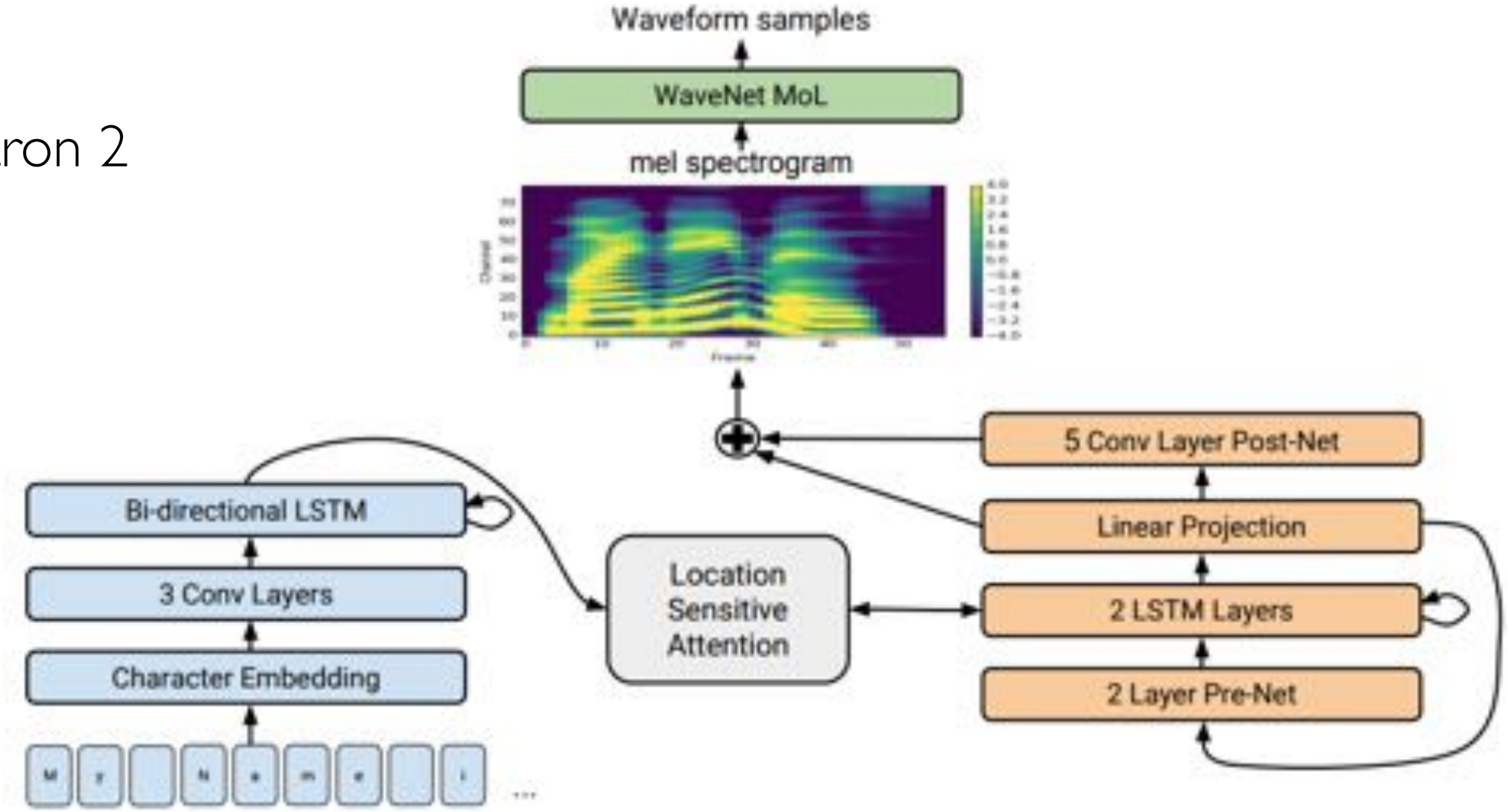
Part 2 — Here comes machine learning

the so-called 'end-to-end' approaches

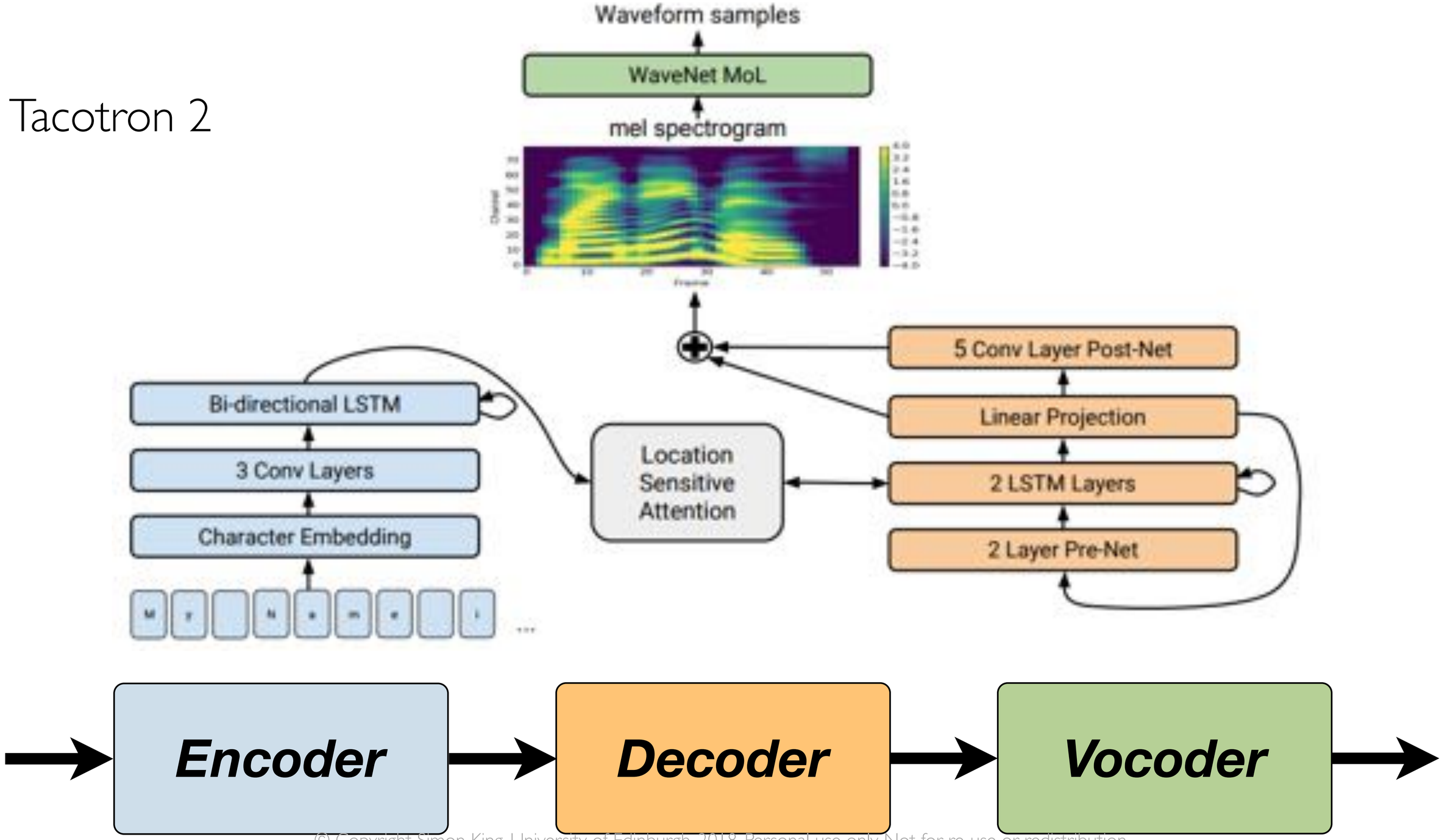
Stages of processing

- feature extraction
- regression
- waveform generation

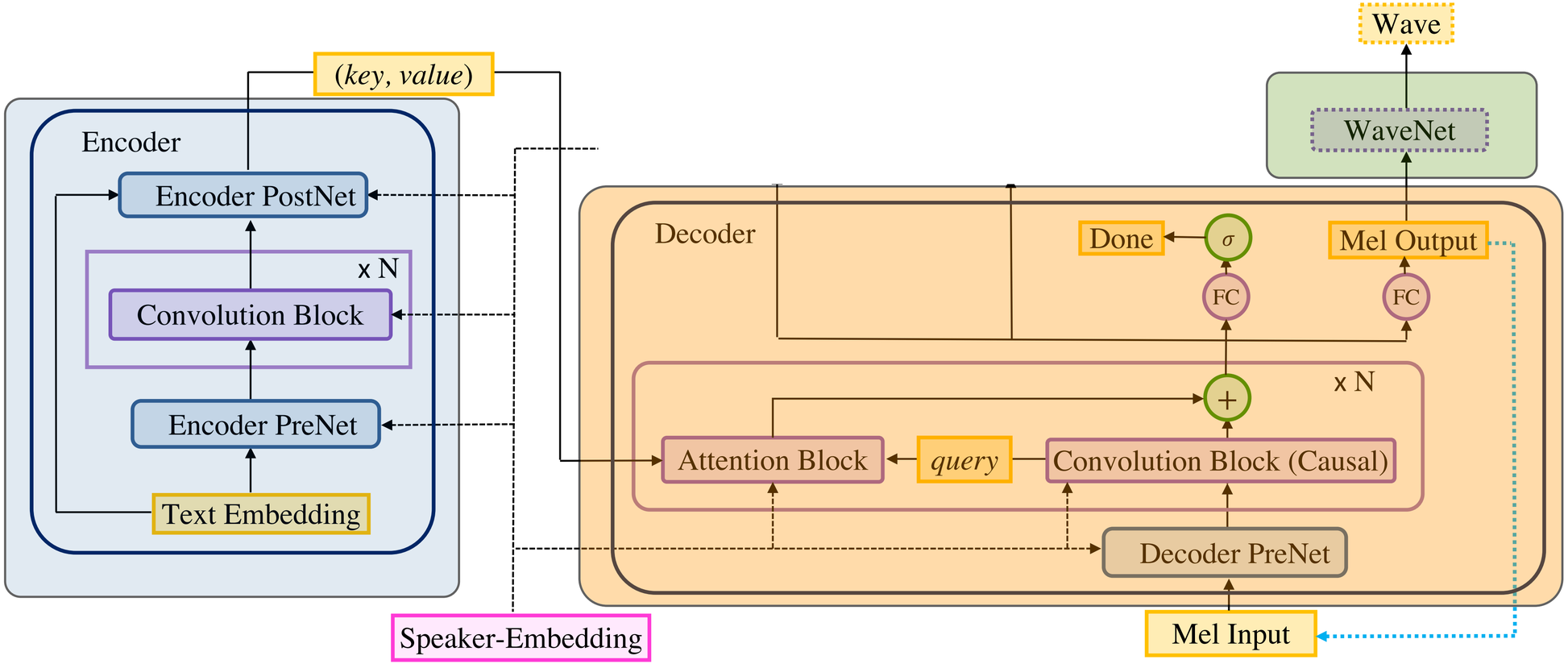
Tacotron 2

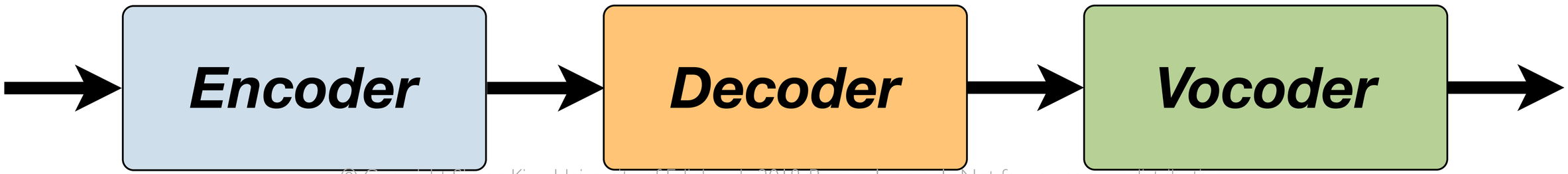


Tacotron 2



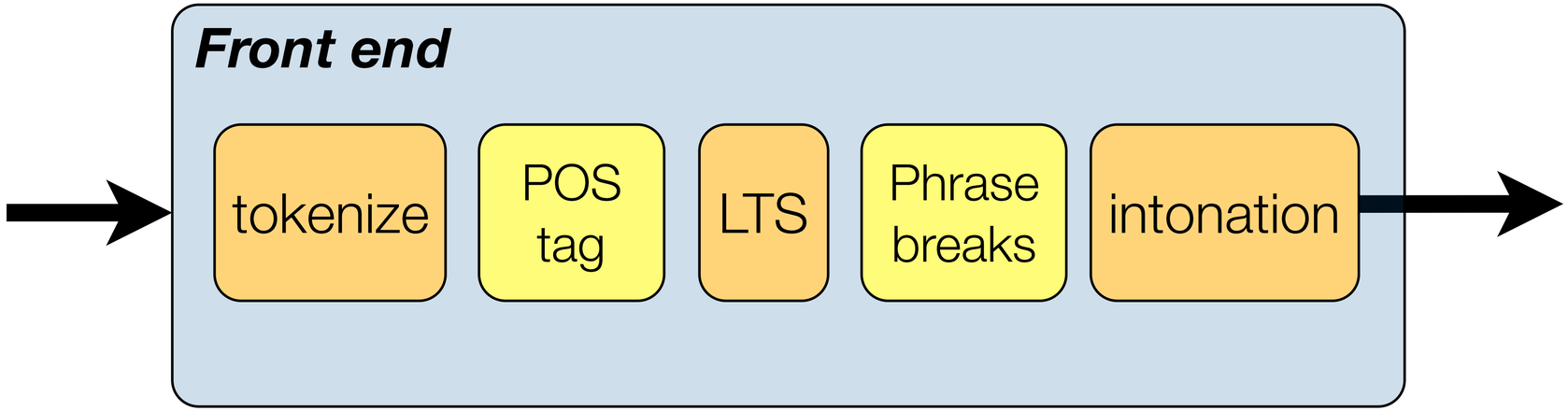
Deep Voice 3





Part 3 — The best of both

Traditional approach



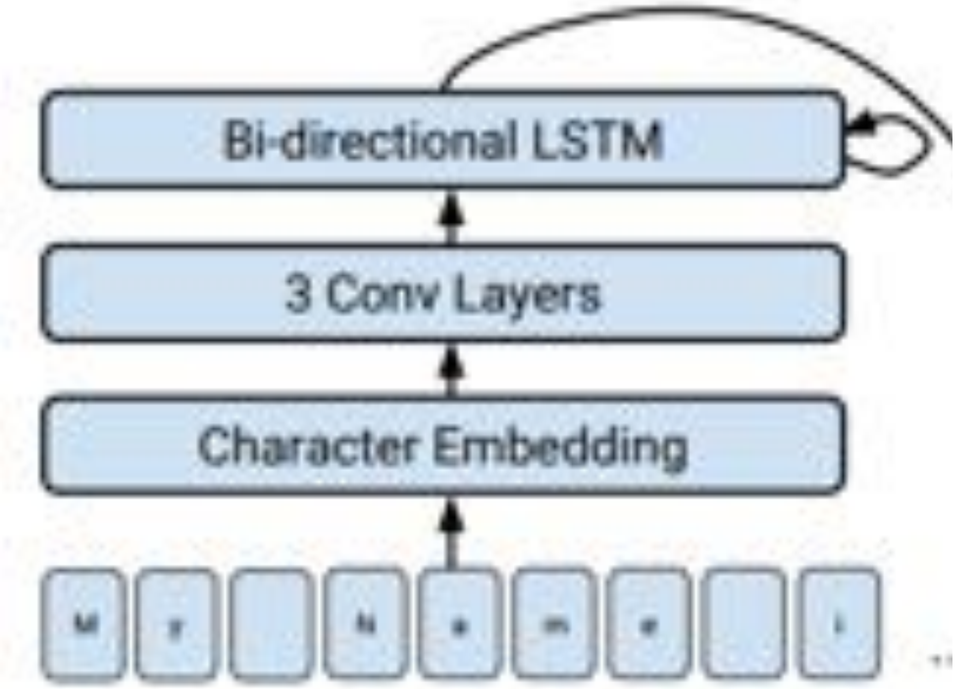
Traditional — explicit pronunciation dictionary + letter-to-sound model

```
ADVOCATING  AE1 D V AH0 K EY2 T IH0 NG
ADVOCATION  AE2 D V AH0 K EY1 SH AH0 N
ADWEEK  AE1 D W IY0 K
ADWELL  AH0 D W EH1 L
ADY  EY1 D IY0
ADZ  AE1 D Z
AE  EY1
AEGEAN  IH0 JH IY1 AH0 N
AEGIS  IY1 JH AH0 S
AEGON  EY1 G AA0 N
AELTUS  AE1 L T AH0 S
AENEAS  AE1 N IY0 AH0 S
AENEID  AH0 N IY1 IH0 D
AEQUITRON  EY1 K W IH0 T R AA0 N
AER  EH1 R
AERIAL  EH1 R IY0 AH0 L
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
```

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

\+ a statistical model learned from this data

New — encoder learns a character sequence embedding



Comparing traditional and new approaches to pronunciation

- The traditional approach
  - explicit representation of pronunciation (syllables + phonemes + lexical stress)
  - write a pronunciation dictionary (not usually speaker-specific)
  - learn to extrapolate from that with a statistical model

- The end-to-end approach
  - annotate a **small** quantity of text with speech using a **single** human talker
    - e.g., 40 hours of speech ≈ 2400 minutes ≈ 360k word tokens
    - which may have about 40k unique word types (random newswire text in English Gigaword)
  - number of word types observed in the speech corpus is substantial
    - so it should be feasible to learn a general pronunciation model from spoken data

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 …

Traditional approach to Non-Standard Words — detect+classify+expand

```
2011 ⇨ NYER   ⇨ twenty eleven
£100 ⇨ MONEY  ⇨ one hundred pounds
IKEA ⇨ ASWD   ⇨ apply letter-to-sound
100  ⇨ NUM    ⇨ one hundred
```

TABLE I. Taxonomy of non-standard words used in hand-tagging and in the text normalization models

	EXPN	abbreviation	*adv, N.Y, mph, gov't*
alpha	LSEQ	letter sequence	*CIA, D.C, CDs*
	ASWD	read as word	*CAT*, proper names
	MSPL	misspelling	*geogaphy*
	NUM	number (cardinal)	*12, 45, 1/2, 0·6*
	NORD	number (ordinal)	*May 7, 3rd, Bill Gates III*
	NTEL	telephone (or part of)	*212 555-4523*
	NDIG	number as digits	*Room 101*
N	NIDE	identifier	*747, 386, I5, pc110, 3A*
U	NADDR	number as street address	*5000 Pennsylvania, 4523 Forbes*
M	NZIP	zip code or PO Box	*91020*
B	NTIME	a (compound) time	*3·20, 11:45*
E	NDATE	a (compound) date	*2/2/99, 14/03/87* (or US) *03/14/87*
R	NYER	year(s)	*1998, 80s, 1900s, 2003*
S	MONEY	money (US or other)	*$3·45, HK$300, Y20,000, $200K*
	BMONEY	money tr/m/billions	*$3·45 billion*
	PRCT	percentage	*75%, 3·4%*
	SPLT	mixed or "split"	*WS99, x220, 2-car* (see also SLNT and PUNC examples)
	SLNT	not spoken, word boundary	word boundary or emphasis character: *M.bath, KENT*RLTY, _really_*
M			
I	PUNC	not spoken,	non-standard punctuation: "***" in
S		phrase boundary	*$99,9K***Whites*, "…" in *DECIDE…Year*
C	FNSP	funny spelling	*sllooooooww, sh*t*
	URL	url, pathname or email	*http://apj.co.uk, /usr/local, phj@tpt.com*
	NONE	should be ignored	ascii art, formatting junk

Sproat et al, "Normalization of non-standard words"
Computer Speech and Language (2001) 15, 287–333
doi:10.1006/csla.2001.0169

A machine-learning approach to Non-Standard Words (arXiv:1611.00068v2)

RNN Approaches to Text Normalization: A Challenge

Richard Sproat, Navdeep Jaitly
Google, Inc.
{rws,ndjaitly}@google.com

Abstract

This paper presents a challenge to the community: given a large corpus of *written* text aligned to its normalized *spoken* form, train an RNN to learn the correct normalization function. We present a data set of general text where the normalizations were generated using an existing text normalization component of a text-to-speech system. This data set will be released open-source in the near future.

1 Introduction

Within the last few years a major shift has taken place in speech and language technology: the field has been taken over by deep learning approaches. For example, at a recent NAACL conference well more than half the papers related in some way to word embeddings or deep or recurrent neural networks.

This change is surely justified by the impressive performance gains to be had by deep learning, something that has been demonstrated in a range of areas from image processing, handwriting recogni-

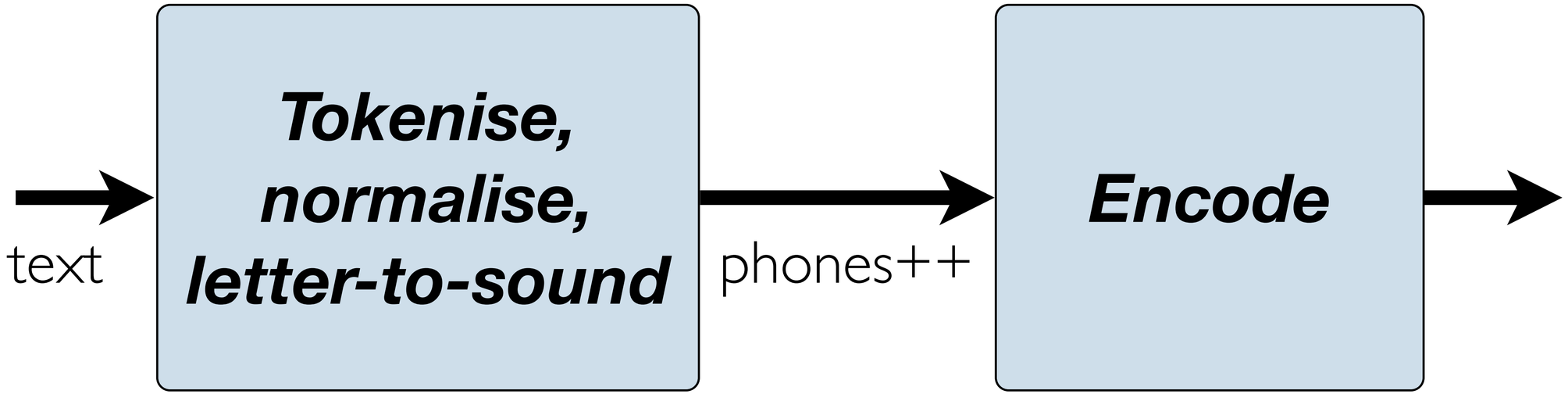
Comparing traditional and new approaches Non-Standard Words (NSWs)

- The traditional approach
  - **explicit** capture of human knowledge
  - annotate a **large** quantity of NSWs with **categories** using **many** human labellers
  - learn an automatic NSW classifier from that data
  - write a specialised expander for each type (simple, deterministic rules are often enough)

- The end-to-end approach *
  - annotate a relatively **small** quantity of text with **speech** using a **single** human talker
  - **implicit** capture of human knowledge

** actually, most end-to-end systems don't even attempt this; they require normalised text*

The best of both



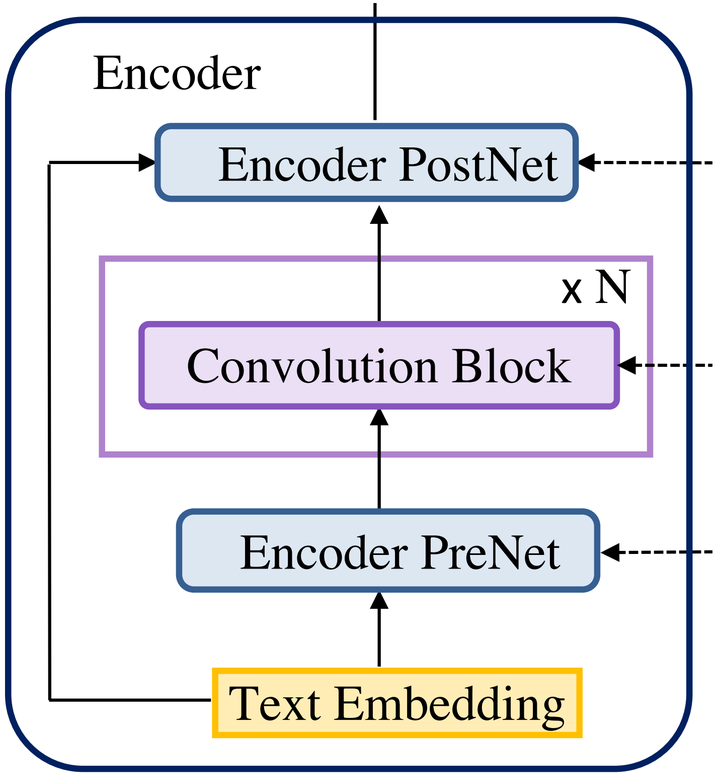
The best of both — some text normalisation
Deep Voice 3 (arXiv:1710.07654v3)

3.1 Text Preprocessing

Text preprocessing is crucial for good performance. Feeding raw text (characters with spacing and punctuation) yields acceptable performance on many utterances. However, some utterances may have mispronunciations of rare words, or may yield skipped words and repeated words. We alleviate these issues by normalizing the input text as follows:

1. We uppercase all characters in the input text.
2. We remove all intermediate punctuation marks.
3. We end every utterance with a period or question mark.
4. We replace spaces between words with special separator characters which indicate the duration of pauses inserted by the speaker between words. We use four different word separators, indicating (i) slurred-together words, (ii) standard pronunciation and space characters, (iii) a short pause between words, and (iv) a long pause between words. For example, the sentence "Either way, you should shoot very slowly," with a long pause after "way" and a short pause after "shoot", would be written as "Either way%you should shoot/very slowly%." with % representing a long pause and / representing a short pause for encoding convenience. [2]

The best of both — characters **and** phonemes
Deep Voice 3 (arXiv:1710.07654v3)

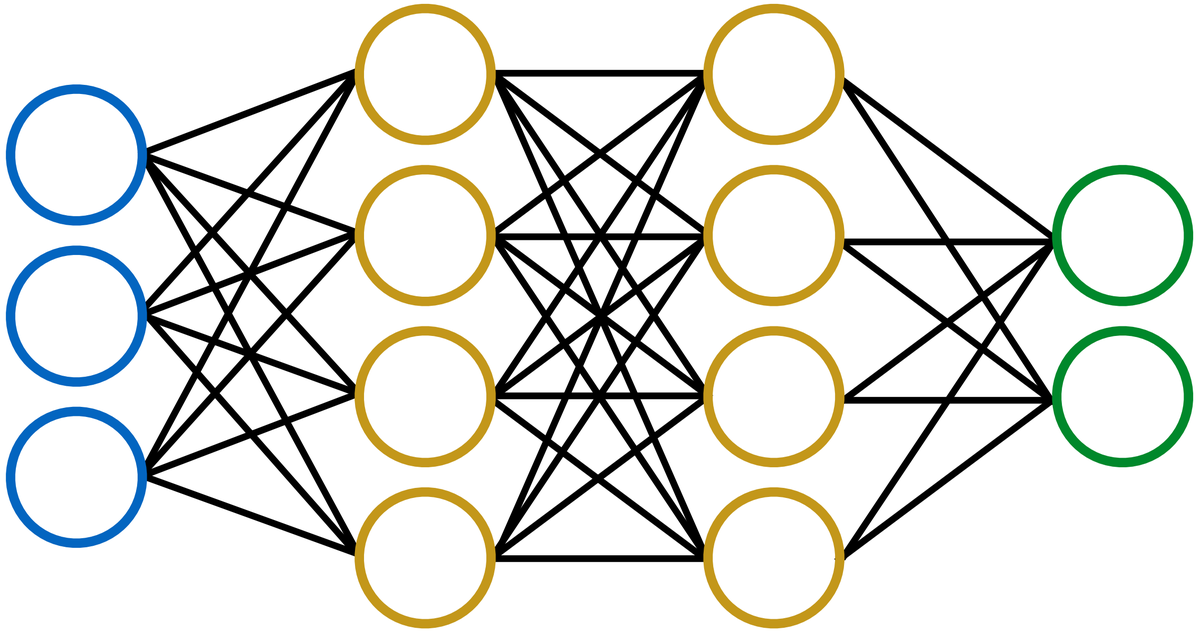


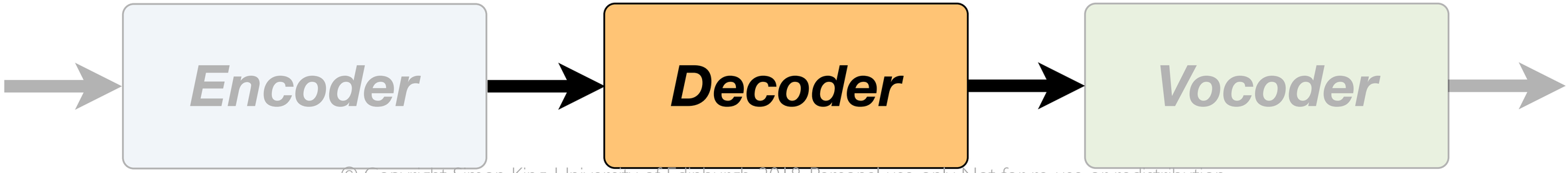
3.2 Joint Representation of Characters and Phonemes

Deployed TTS systems (e.g., Capes et al., 2017; Gonzalvo et al., 2016) should include a way to modify pronunciations to correct common mistakes (which typically involve proper nouns, foreign words, and domain-specific jargon). A conventional way to do this is to maintain a dictionary to map words to their phonetic representations.

Our model can directly convert characters (including punctuation and spacing) to acoustic features, and hence learns an *implicit* grapheme-to-phoneme model. This implicit conversion is difficult to correct when the model makes mistakes. Thus, in addition to character models, we also train phoneme-only models and mixed character-and-phoneme models by allowing phoneme input option explicitly. These models are identical to character-only models, except that the input layer of the encoder sometimes receives phoneme and phoneme stress embeddings instead of character embeddings.

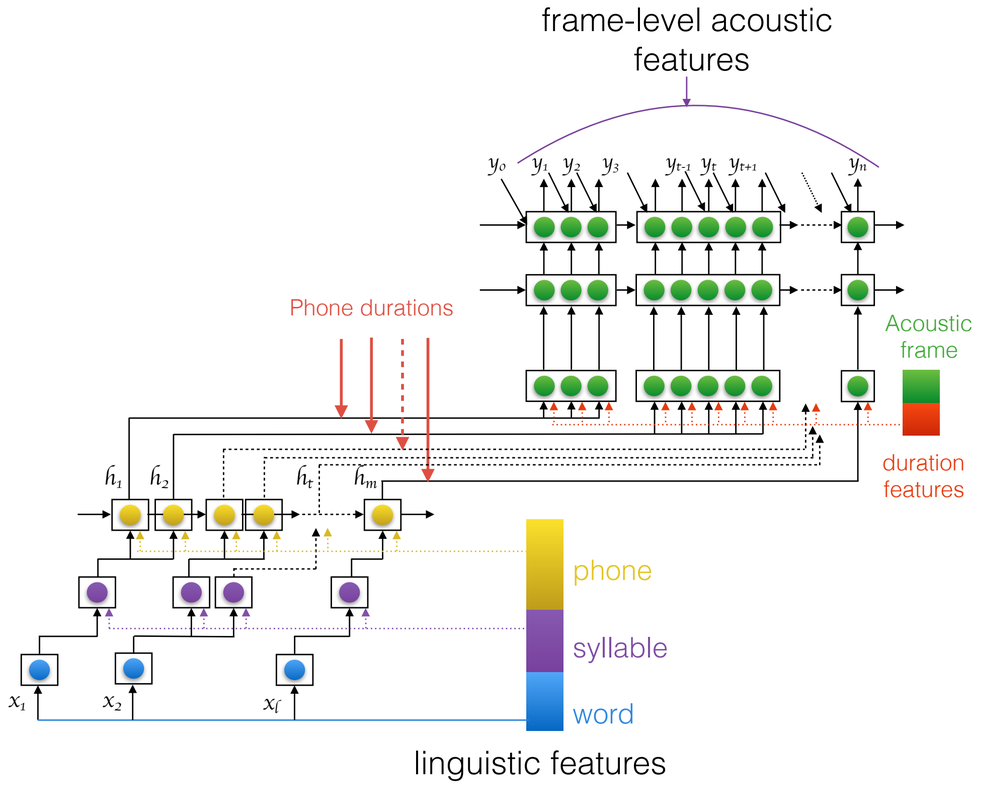
Traditional decoder operates frame-by-frame on upsampled linguistic features

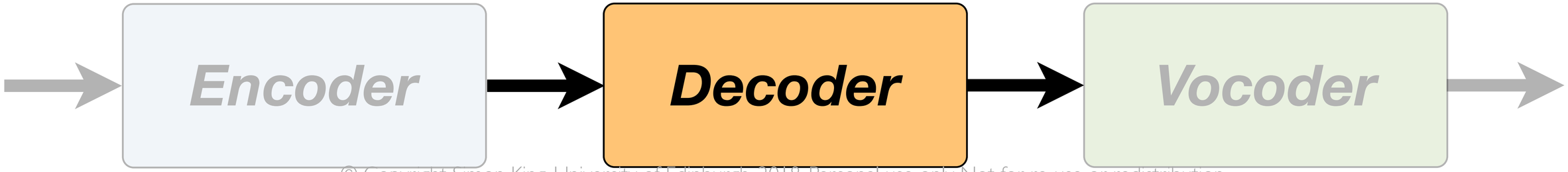




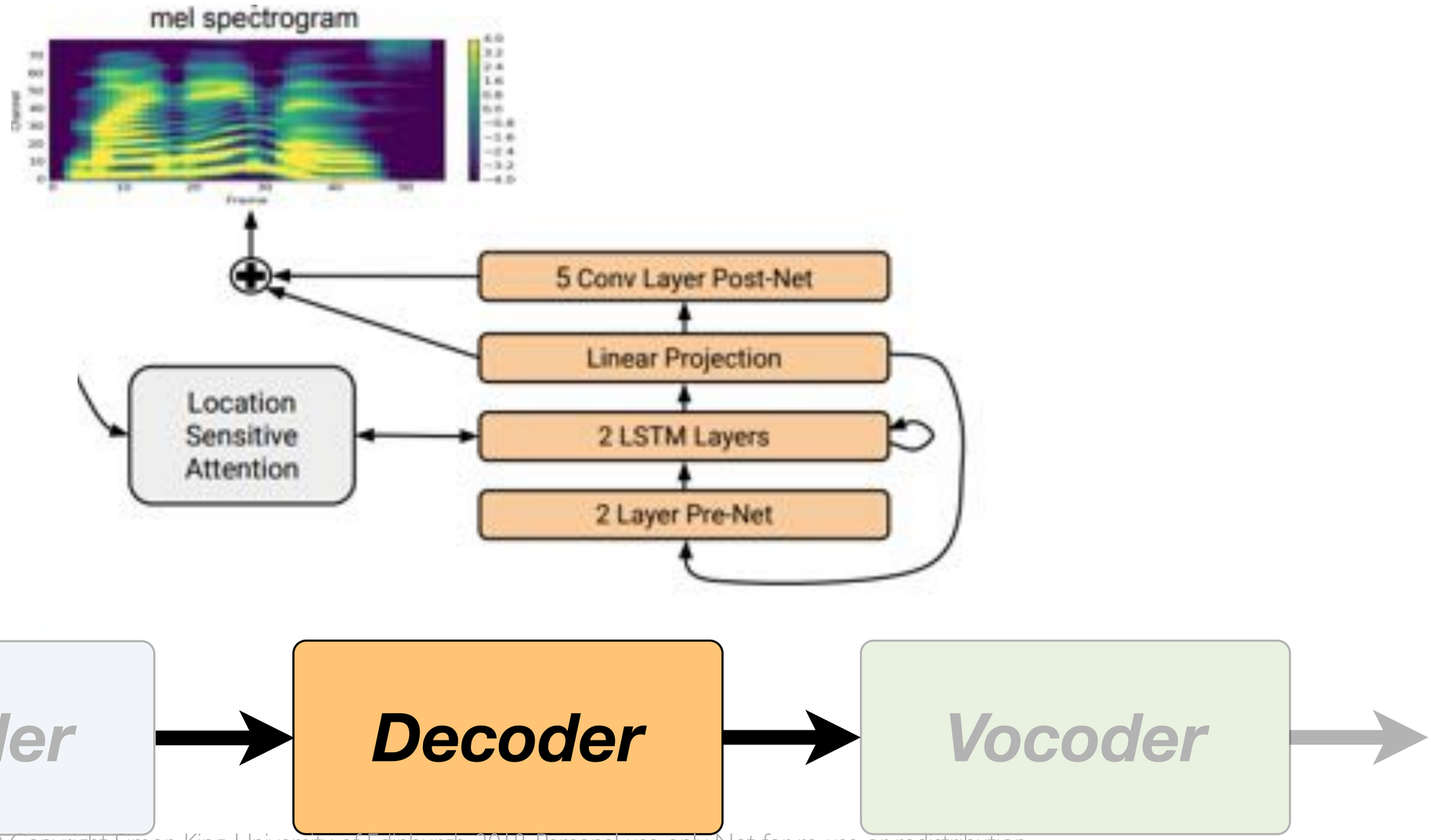
New decoders that bridge linguistic and acoustic timescales

Ronanki, Watts & King, Interspeech 2017

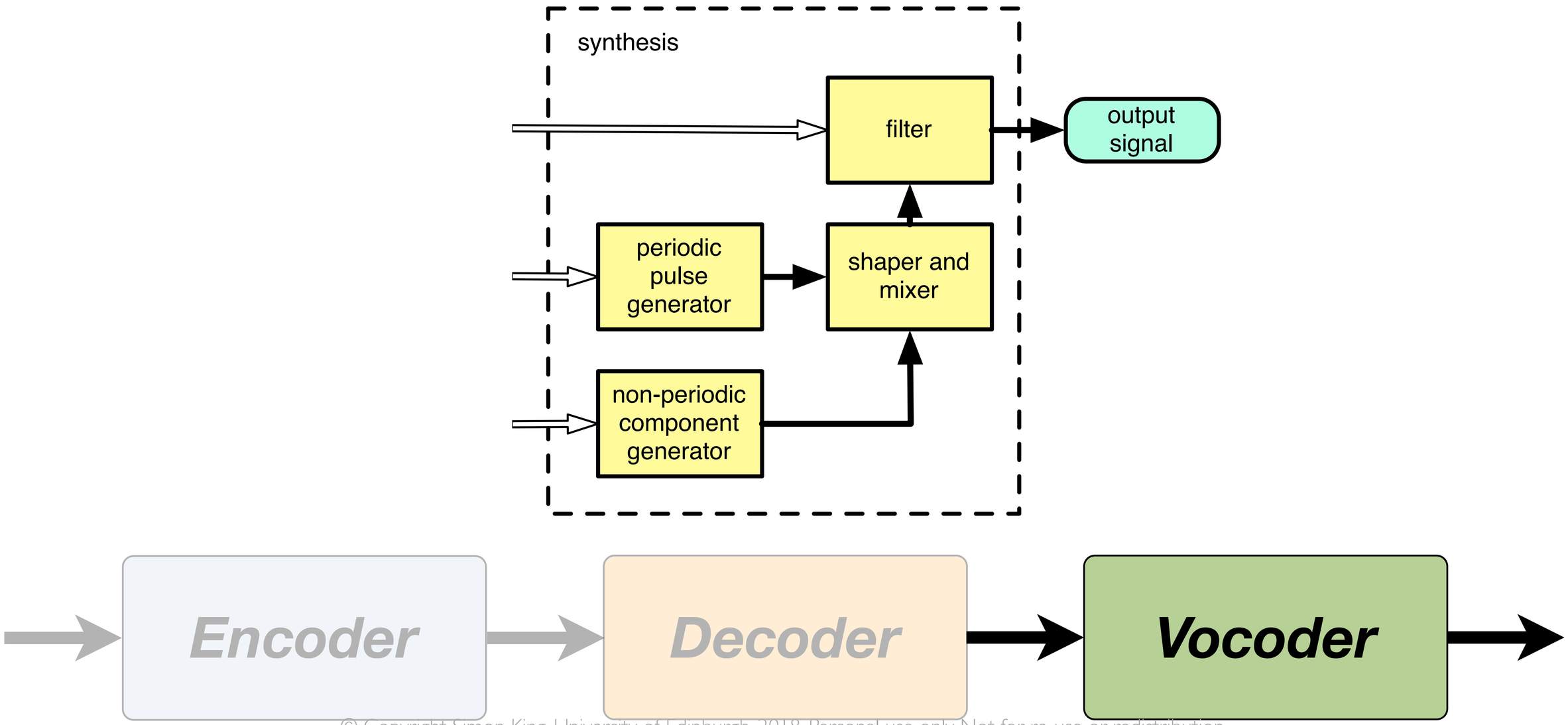




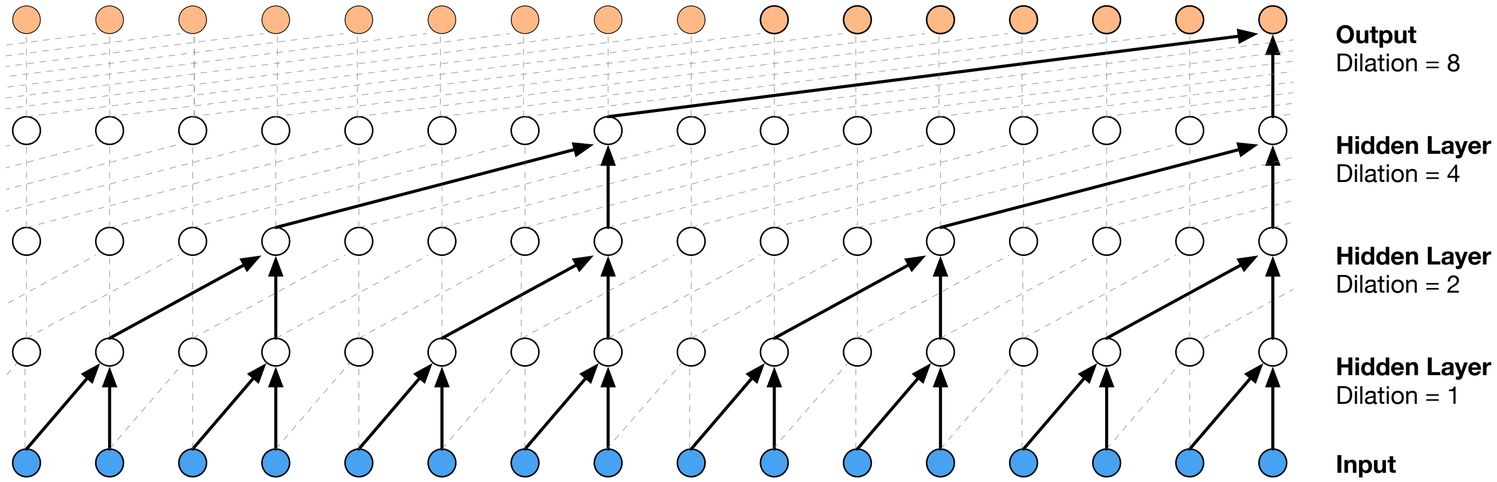
New decoders are true sequence models — Tacotron 2



Traditional vocoders use carefully crafted signal processing



New vocoders are learned from data
but contain sub-components that mimic traditional approaches



WAVENET: A GENERATIVE MODEL FOR RAW AUDIO

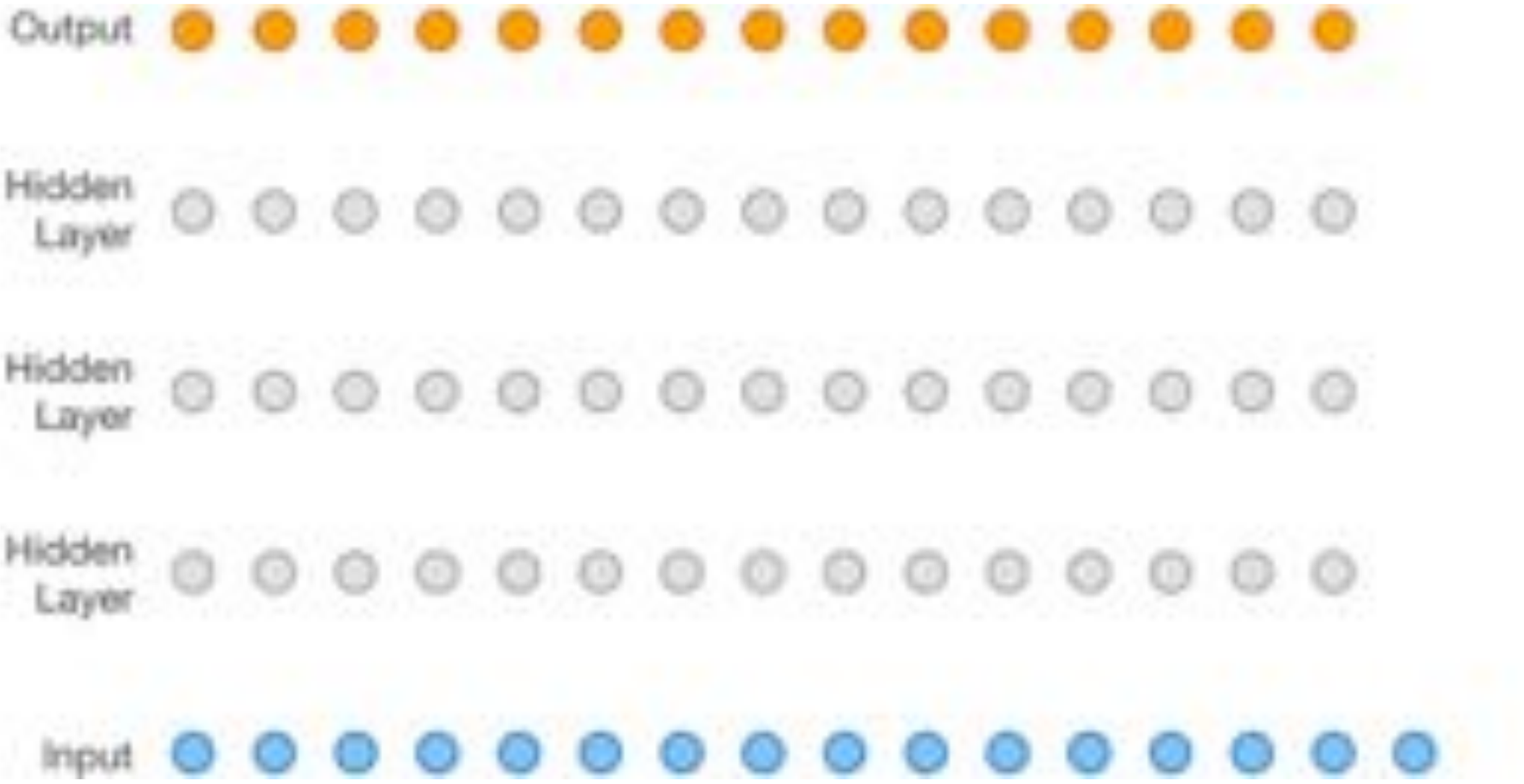
Aäron van den Oord **Sander Dieleman** **Heiga Zen**[†]

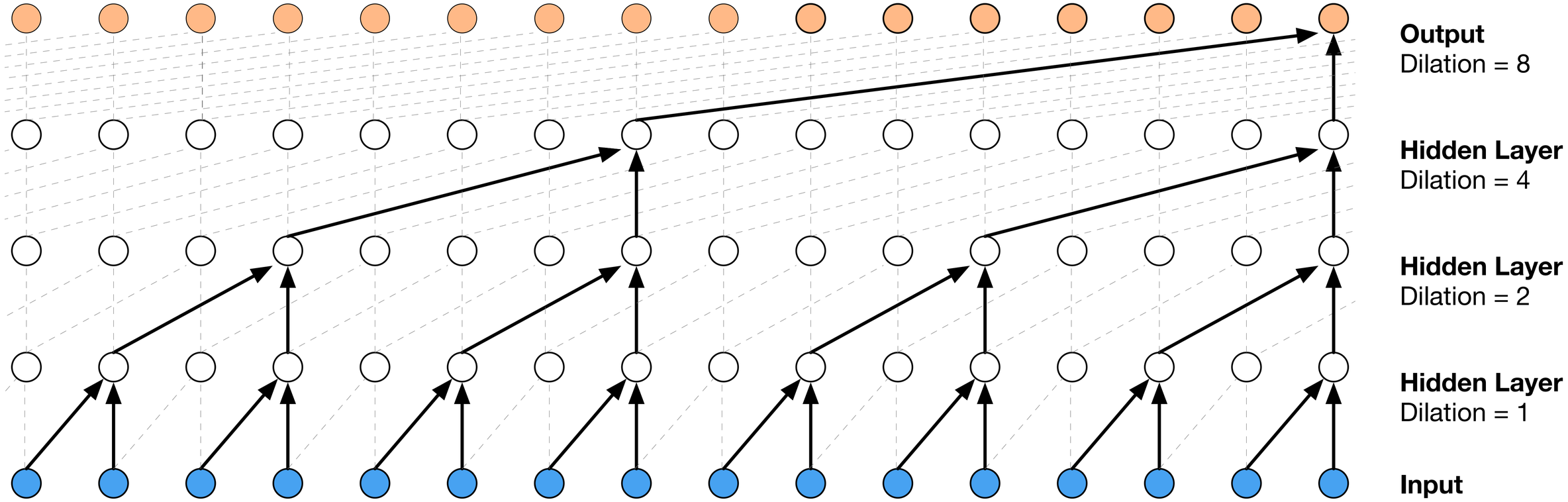
Karen Simonyan **Oriol Vinyals** **Alex Graves**

Nal Kalchbrenner **Andrew Senior** **Koray Kavukcuoglu**

{avdnoord, sedielem, heigazen, simonyan, vinyals, gravesa, nalk, andrewsenior, korayk}@google.com
Google DeepMind, London, UK
[†] Google, London, UK

ABSTRACT

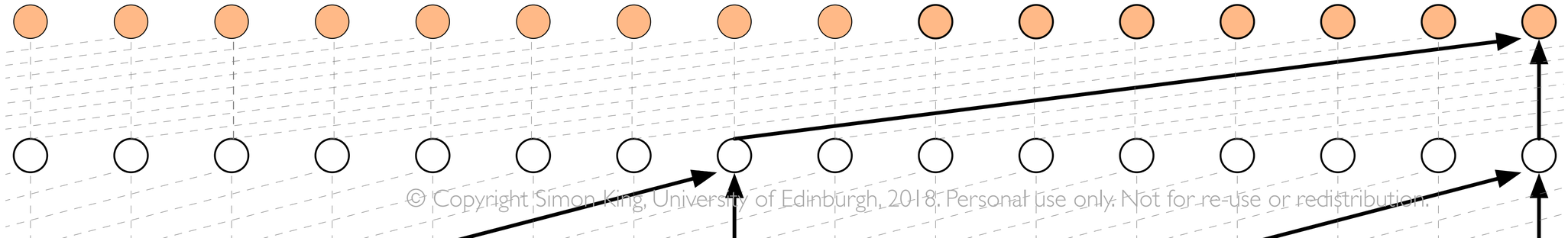




“one-hot” coding of 8 bit quantised waveform sample = **1-of-256**

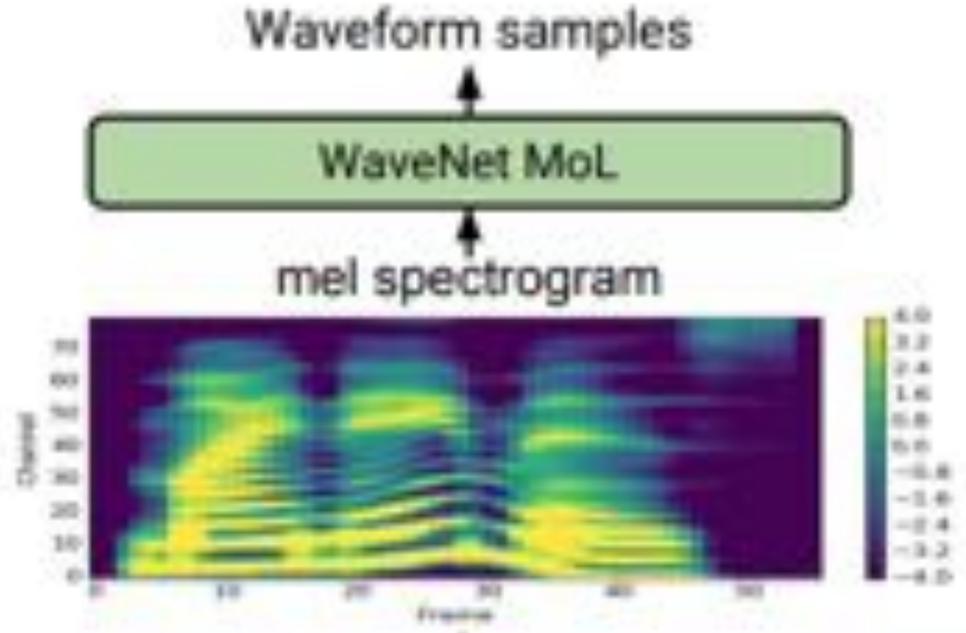
0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0

(in reality, many more layers of model go here)



The best of both

Wavenet conditioned not on text, but **on spectrogram**



Are you traditional, purist, or pragmatic ?

- Traditional methods learn from a **wide variety of data**
  - but there are many inconsistencies between these data sources
    - e.g., dictionary does not match the speaker's accent

- Purist end-to-end approaches learn *only* from **parallel text + speech data**
  - but they cannot make use of additional text-only (or speech-only) data

- "Best of both" approaches take a *pragmatic* view
  - traditional approaches to normalise the text into "phones++"
  - end-to-end learning to regress from that to the spectrogram
  - neural vocoder to generate the waveform, conditioned on that spectrogram

Opportunities

(1) Traditional text processing works well

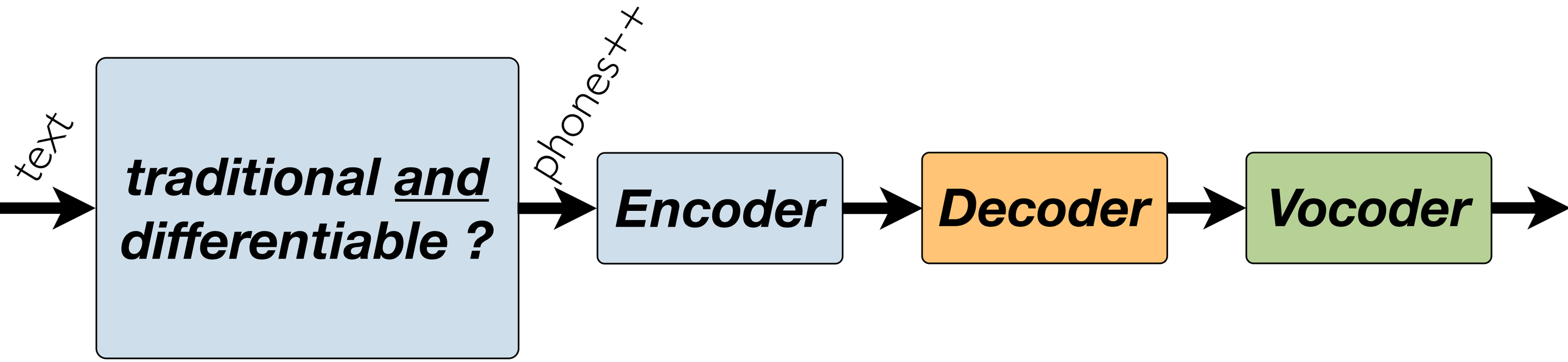
- unfortunately, these methods are not **differentiable**
- and therefore not learnable end-to-end

(2) Even the best end-to-end models still use flat input sequences

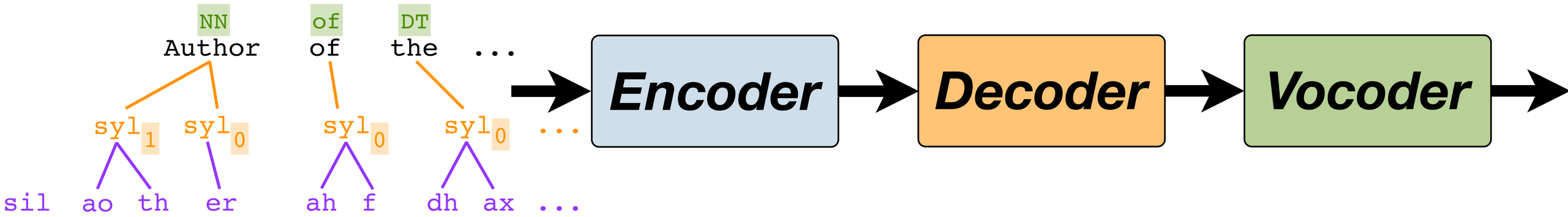
- but linguistic **structures** are *not flat*
- e.g., phrases - words - syllables - phones

(3) Even in end-to-end approaches, we have many choices about what we **optimise**

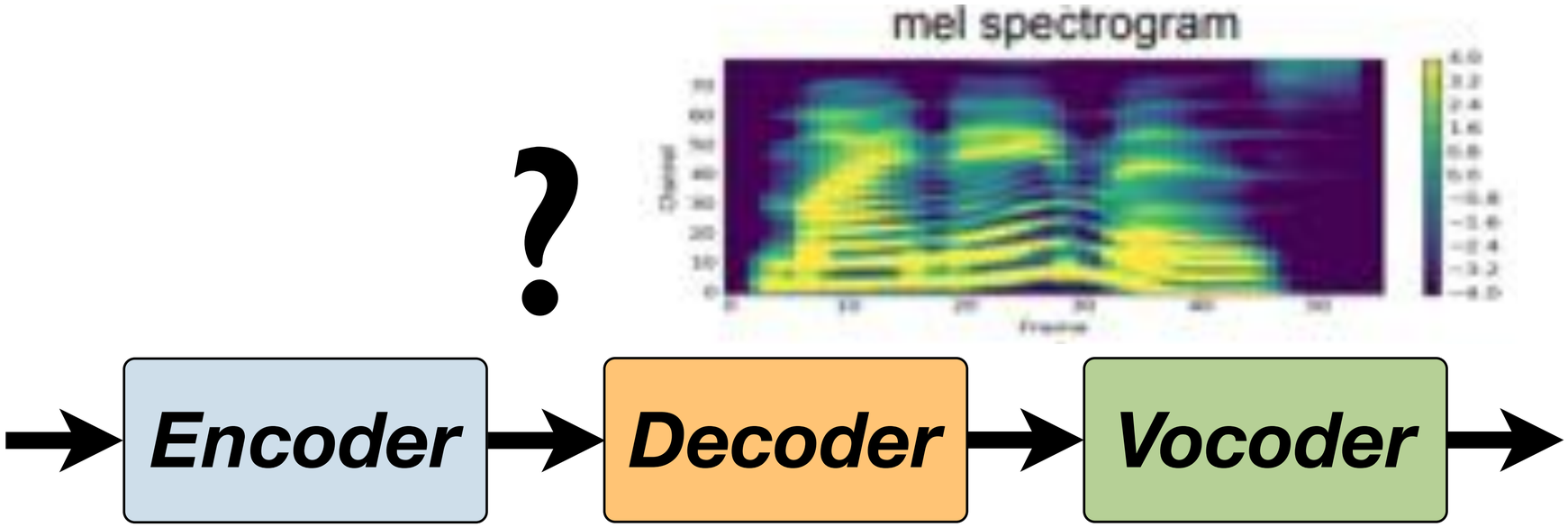
Opportunity — differentiable traditional approaches

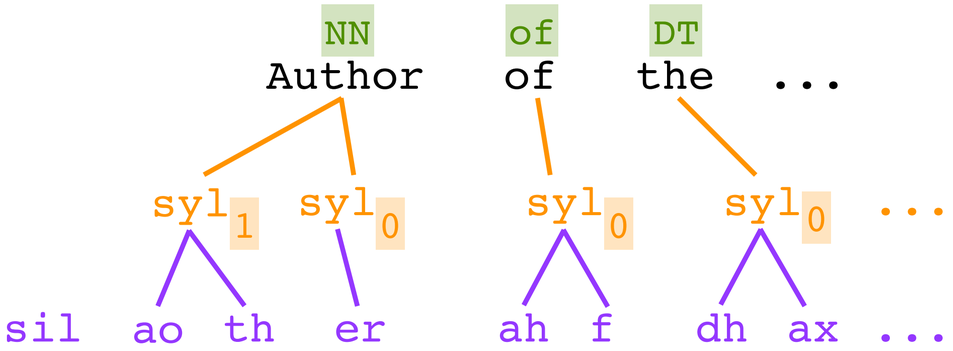


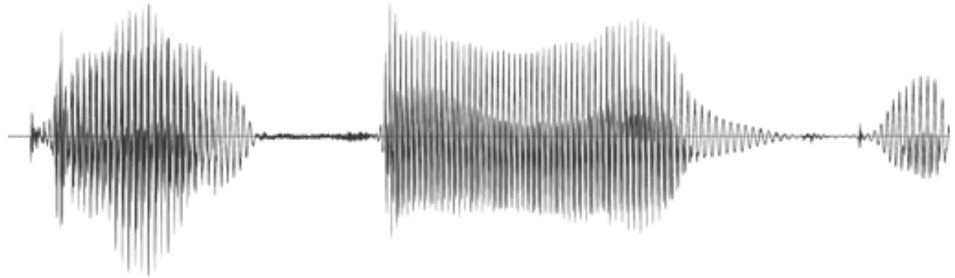
Opportunity — make use of linguistic structure



Opportunity — choose what to optimise







What we **are** optimising vs. what do we **want** to optimise

- explicit intermediate representations imply minimisation of error in that domain
  - e.g., regression model minimises error in the **spectrogram** domain

- intermediate representations are therefore a critical design choice, at every stage
  - so, if we care about minimising **pronunciation errors**, perhaps an explicit representation of pronunciation is not a bad idea
  - or, if we wish to minimised **perceived error** in the speech output, then a perceptually-relevant representation would be nice (the log Mel spectrogram is on the right lines, but too simplistic)

CSTR website: **www.cstr.ed.ac.uk**

Teaching website: **speech.zone**