



# Speech synthesis

Chloé Clavel, INRIA Paris



# Speech synthesis or TTS (Text To Speech)

## Objective :

- be able to read any written text

### Automatic speech recognition (ASR)

—  → "OK Google, directions home"

### Text-to-speech synthesis (TTS)

"Take the first left" → — 

# Speech synthesis or TTS (Text To Speech)

## Purpose different from talking machines

- Talking machine = concatenation of word/phrase records
- Ex: Pre-recorded voices from the metro or talking clocks

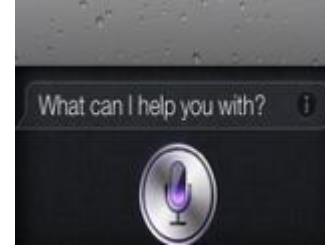
“au quatrième top il sera exactement :

(nombre inséré) heure (nombre inséré) minutes (nombre inséré) secondes

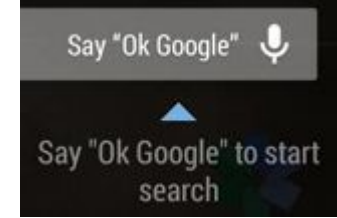
- Constraints :
  - limited vocabulary,
  - sentence with fixed structures,
  - reasonable size of the record base

# Examples of applications

- Telecommunications services
- Cinema, traffic and bank account information
- Reading of SMS, emails for blind people
- Learning foreign languages
- Oral conversation with an animated conversational agent (open source software: Open Mary)
- Virtual Assistants
- Companion robots



Virtual assistant



# Corpora

**Corpora for ASR:** large corpora with thousands of hours of speech from many speakers

-> ASR systems need to be speaker-independent (needs to generalize well to an unseen speaker)

**Corpora for TTS:** much less data but all for one speaker

-> TTS systems are generally speaker-dependent : trained to have a consistent voice

Ex: LJ speech corpus : 24 hours of one speaker reading audio books (Ito and Johnson, 2017)

# Production of the voice signal: the ancestors

Von Kempelen (1791)

Manual voice synthesizer

Reproduction of the human vocal tract

Production of vocal sounds

Ex : "nostrils" (narines) for making the "  
"m" sounds

levers (leviers) and tubes dedicated to '  
"sh" sounds

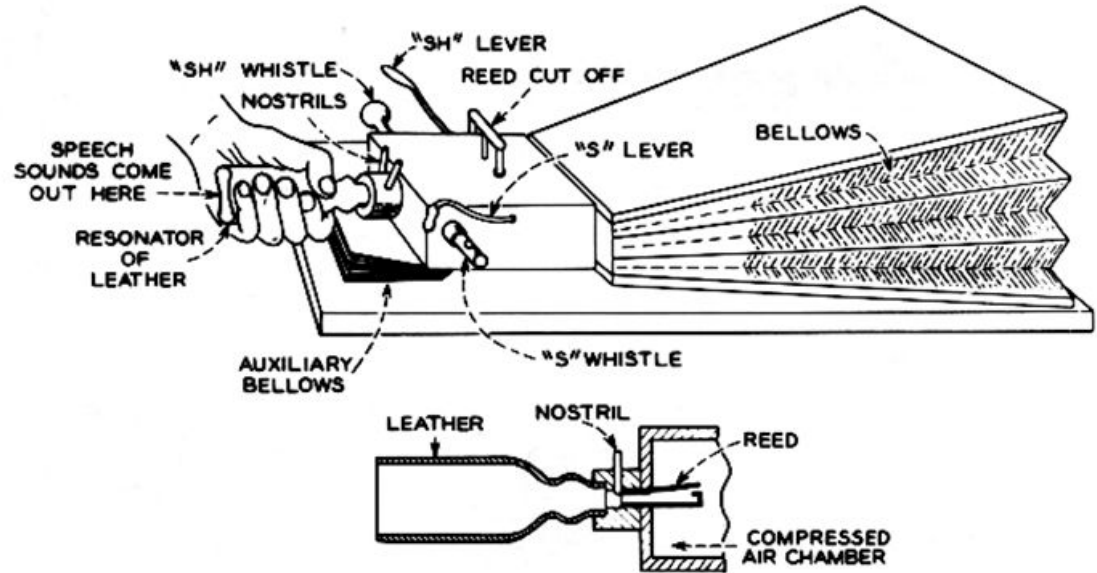
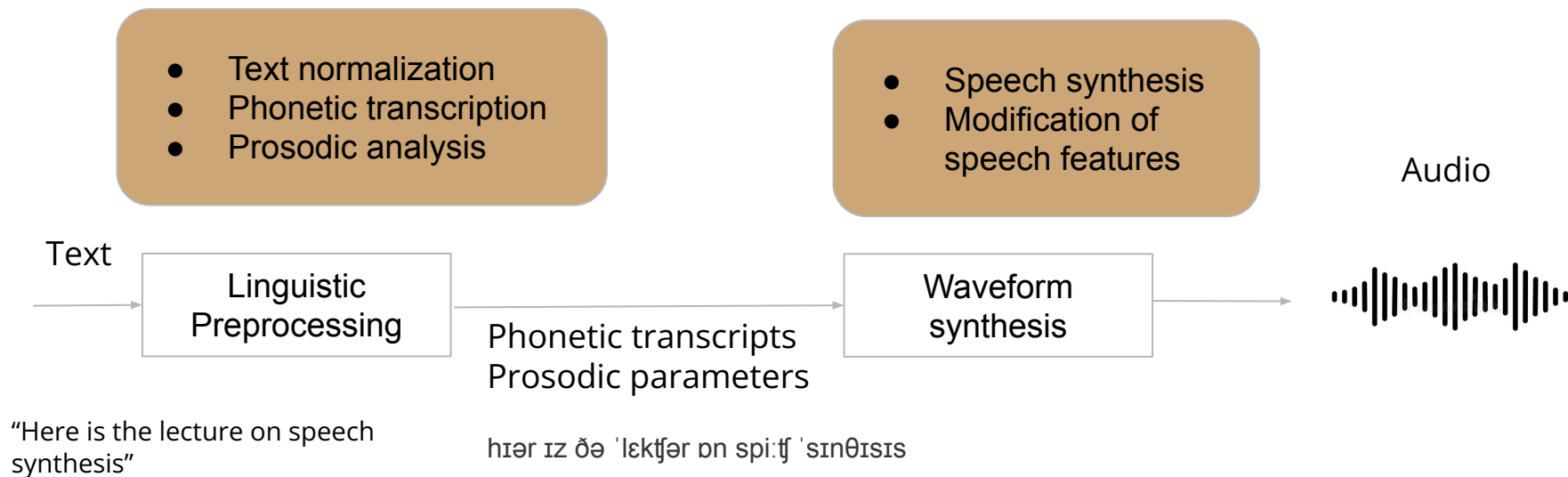


FIG. 10. Wheatstone's reconstruction of von Kempelen's speaking machine.<sup>1</sup>

# Overall architecture of a two-block TTS



# Analogy with the mode of speech production

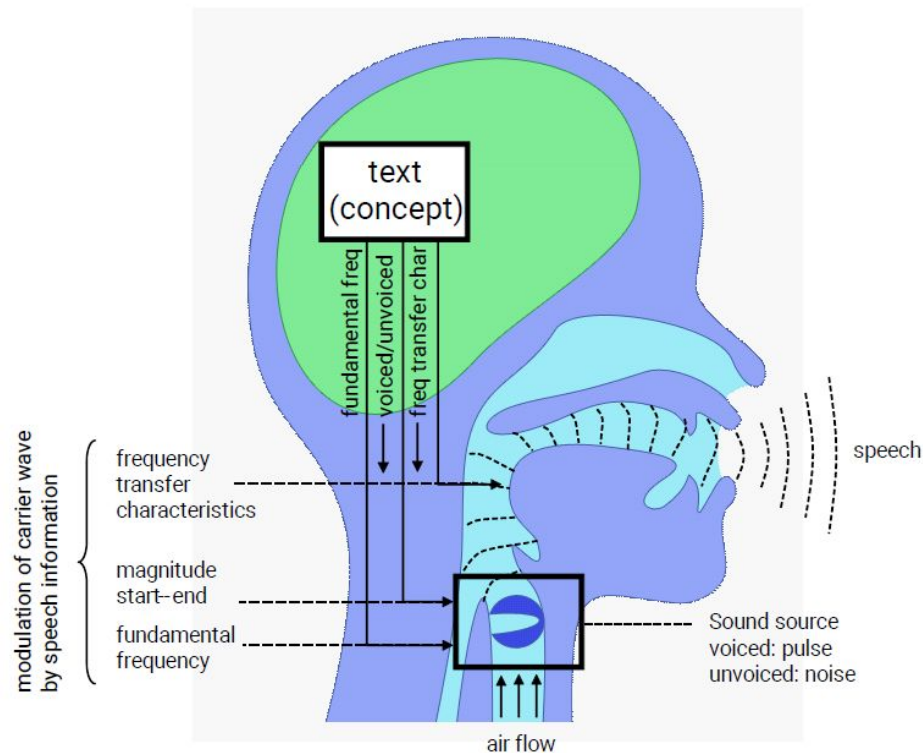


Figure from [Generative Model-Based Text-to-Speech Synthesis](#)  
[Andrew Senior \(DeepMind London\)](#)




# Different approaches

1. Rule-Based Approaches [Klatt, 1980]
2. Synthesis by concatenation [Hunt & Black, 1996].
3. Parametric synthesis:
  - generative synthesis from models (HMM) [Zen et al., 2009].
  - contents and characteristics of the speech signal can be controlled via the inputs to the model
4. Deep learning approaches (WaveNet) [Van Den Oord et al., 2016].


Examples :

<https://deepmind.com/blog/article/wavenet-generative-model-raw-audio>



Linguistic processing :

- 1/Text normalization
- 2/grapheme-phoneme conversion



# Linguistic processing

Shared steps by all the different approaches :

- Rule-Based Approaches [Klatt, 1980], Synthesis by concatenation [Hunt & Black, 1996], Parametric synthesis [Zen et al., 2009].
  - Text -> Phonetic transcripts + prosodic parameters
- Deep learning approaches (WaveNet) [Van Den Oord et al., 2016] :
  - Text -> sequence of linguistic and phonetic features (which contain information about the current phoneme, syllable, word, etc.) fed into WaveNet.

# Example: mail reading

Voice synthesis of a mail for the blind or by a virtual assistant

How can we reproduce the process used by humans to read an email?

## 1. Mail structure analysis

- **Segmentation** : Head/Message body (texts, inclusions, non textual segments), Attachments/ Signature
- **How ?** using sequential machine learning approaches (ex: Hidden Markov Models, Recurrent Neural Networks)
- **Difficulties**: Management of tags, marks and typographical variations (indents, dashes, blanks, tabs...)

## 2. Text normalization :

- Literal spelling reformulation (verbalization) or filtering of symbols to be pronounced depending on the context

They live at 224 Mission St. -> They live at two twenty four Mission Street

En % -> en pourcentage

05/01/2023 -> cinq janvier deux-mille-vingt-trois or premier mai deux-mille-vingt-trois)

# Approaches for text normalization

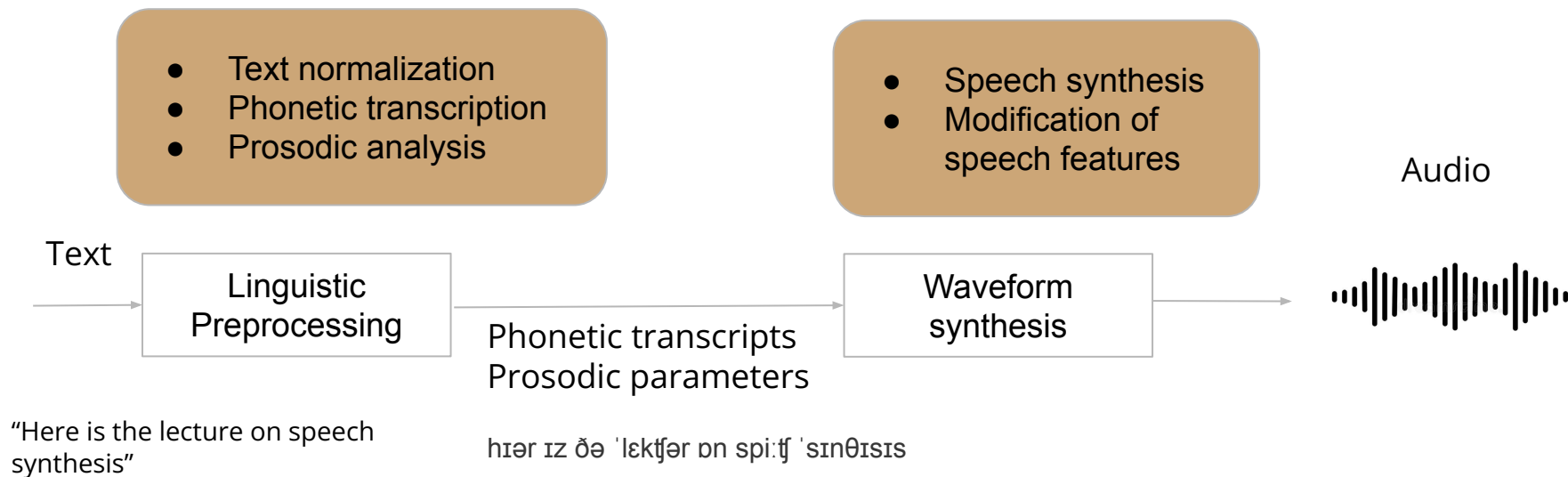
Different types of approaches:

- Named entity recognition systems, dictionary, and rules to detect non-standard words
  - Ex: regular expressions for detecting years `/(1[89][0-9][0-9])|(20[0-9][0-9]/`
- Rules to produce the verbalization
  - Ex: the Kestrel system (Ebden and Sproat, 2015) based on a verbalization grammar
- Encoder-decoder models:
  - expert-labeled training sets with the mapping between original sequence and normalized sequence (similar problem as **machine translation**)
    - Ex: [Sproat and Gorman 2018](#), [Zhang et al. 2019](#).
  - Verbalization grammars can be used to constraint the decoding step.

Practice : test the ability to deal with problematic character sequence

<https://www.acapela-group.com/fr/demos/>

# Overall architecture of a two-block TTS



# Phonetic transcription

(Normalized) orthographic text



phonetic text

(sequence of phonemes)

ex : /vwav/)

## Voyelles

[a] pas  
[ɑ] pâte  
[e] blé  
[ɛ] bête, lait  
[i] fil  
[ɔ] sol  
[o] beau, do  
[u] trou  
[y] mur  
[ø] bleu  
[œ] fleur  
[ə] renaître

[ɛ̃] pain, fin  
[ɑ̃] blanc  
[ɔ̃] mont  
[œ̃] parfum

## Consonnes

[p] plein  
[b] bois  
[d] dent  
[t] tige  
[k] clair, kiwi  
[g] gare  
[f] fille, éléphant  
[s] sac, bosse  
[ʃ] chameau  
[v] vert  
[z] zèbre  
[ʒ] jeune  
[l] larme  
[R] route  
[m] mode  
[n] note  
[̃] campagne  
[ŋ] jogging

## Semi-consonnes

[j] yo-yo  
[ɥ] cuit  
[w] oui  
[œR] heure  
[waR] victoire

Phonetic  
alphabet in  
French

# Phoneme, Phone : writing the sound form

In French: 36 phonemes : 17 consonants, 16 vowels et 3 semi-vowels

A single **phoneme** (definition seen in 1st lecture) can be given multiple acoustic realizations (e.g., depending on the speaker or on the accentuation/emphasis)

Ex : /œ/

A class of particular implementations of the same phoneme is called a **phone**.



# Phonetic transcription

Transcription +/- difficult depending on the language and complex for French

No bijection between letter space and phoneme space

Ex: A sequence of letters can have several pronunciations:

'ch' can be pronounced:

- /k/ in *chlore*

- /ʃ/ in *château*

# Transcription of isolated words : difficulties

Example : complex rules for the transcription of "oiseau" -> "/wazo/" :

1/ "oi" is transcribed by /wa/, because:

- preceded by a word separator (space)
- not followed by the string "gn" as in "oignon", or by an "n" as in "oindre".

2/ "s" is transcribed by /z/ because

- surrounded by two vowels
- "oiseau" does not belong to an exception list for this rule (ex: "paraSol" ou "vraiSemblance").

3/ "eau" is transcribed /o/, context-independent

# Transcription of isolated words: how does it work?

**Use of lexicons and morphological analysis to define pronunciation rules:**

s -> z / V \_\_ V (in French, ex : oiseau)

Rule that does not work for *a+social* => Exception rule

s -> s / #a,#anti,#pro... \_\_ V

**OR Use of machine learning approaches on corpora :**

observation: orthographic symbol

class : phonetic symbol

descriptors : the context orthographic window around the symbol.

Training samples : pronunciation dictionary

# Transcribe words in context - syntax dependance

**Homographs-heterophones** : words with the same spelling but with a different pronunciations depending on their grammatical function in the sentence

est-V vs est-N,  
couvent-V vs. couvent-N,  
bus-V vs. bus-N,  
violent-V vs. violent-A,  
portions-V vs. portions-N,  
fils-N vs. fils-N....

Require a **syntactic analysis** of the sentence in order to disambiguate the pronunciation

# Transcribe words in context

**Syntactic analysis** : reduce the number of possible lexical categories for each word according to its neighbours.

Ex: one word -> several lexical categories (50% of the occurrences)

(2) La/DET/N/PRO

(3) couvent/N/V

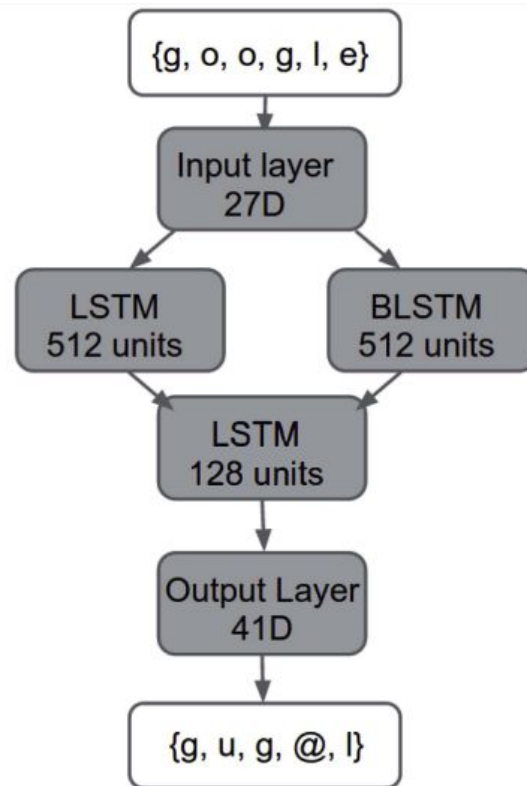
A sentence of 20 words has 210 possible analyses

# Machine learning for phonetic transcription

Inputs : one-hot vectors (size = number of distinct graphemes (here 20))

Output: one-hot vectors (size = number of distinct phonemes) (here 40)

*Grapheme-to-phoneme conversion using long short-term memory recurrent neural networks. ICASSP 2015*



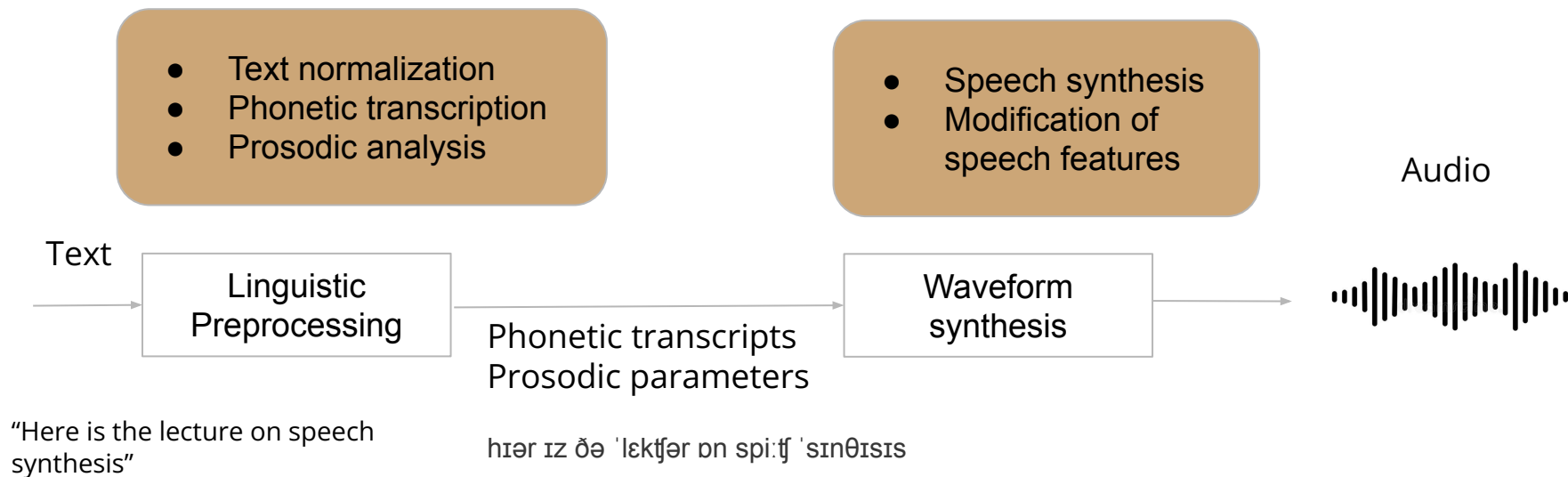


# Linguistic processing

## - prosodic module-



# Overall architecture of a two-block TTS





# Prosodic module : objectives

Calculate prosodic parameters (also called intonation symbols) automatically from the text

*Supra-segmental* speech characteristics :

Melody, Rhythm, Accent and Emphasis

Three main parameters:

- Intonation: variations in the fundamental frequency...-> Melody
- Duration: Segment and pause durations -> Duration
- Intensity: a function of energy -> Melody, Accent and Emphasis

# Extraction of 3 types of symbolic information from the text :

1/Modality of the statement (declarative, imperative, interrogative)

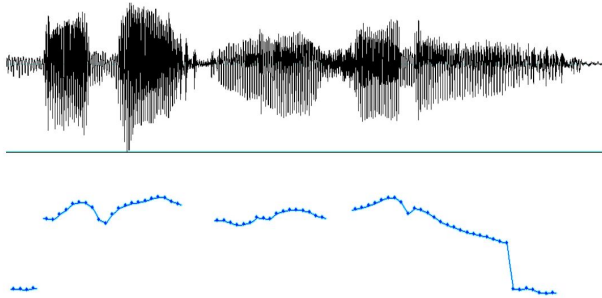
2/Identification of Prosodic Groups

3/Identification of accented syllables within prosodic groups -> duration, energy, micro-melody, rhythm.

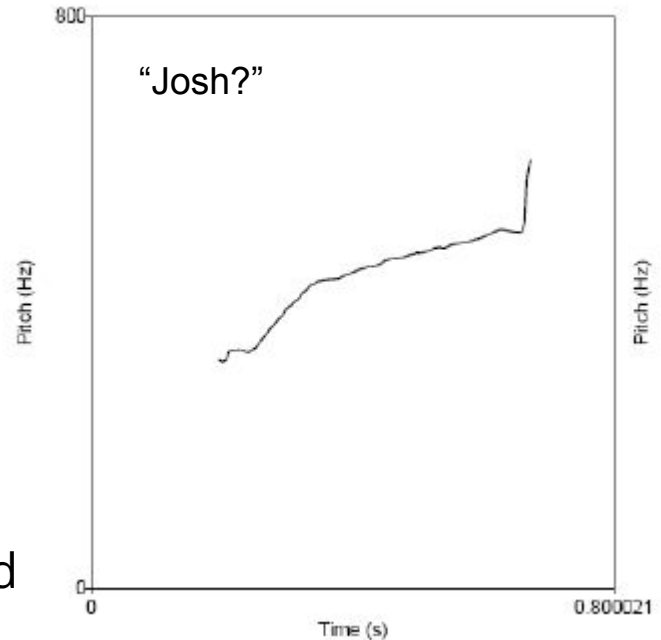
# Extraction of 3 types of symbolic information from the text :

1/Modality of the statement (declarative, imperative, interrogative) -> overall shape of the intonative curve

Ex : Declination: the melodic line generally decreases from the beginning to the end of a declarative sentence.



Method : supervised machine learning on labelled corpora



# Extraction of 4 types of symbolic information from the text :

2/Identification of Prosodic Groups -> position of accents and pauses within the sentence, (re)initialization of the declination curve

Demarcative function of the prosody : from the sentence structure (dot, coma, chunk), we can decide where to put the pauses

Example of tricky cases : « Jacqueline (entend (le bruit de la fenêtre)) » vs. « Jacqueline (entend (le bruit) (de la fenêtre)) »

Methods:

- rules/heuristics
- morpho-syntactic analysis for sentence segmentation
- machine learning approaches on corpora labelled in pauses

### 3/Identification of accented syllables within prosodic groups -> duration, energy, micro-melody, rhythm.

Identification of the accented syllables, the type of the accent

-> determination of the corresponding prosodic parameters

Ex : the emphatic accent is expressed either by:

- an increased strength and duration of the consonant ("la garçon").

- a glottis stroke

- A higher melodic rise

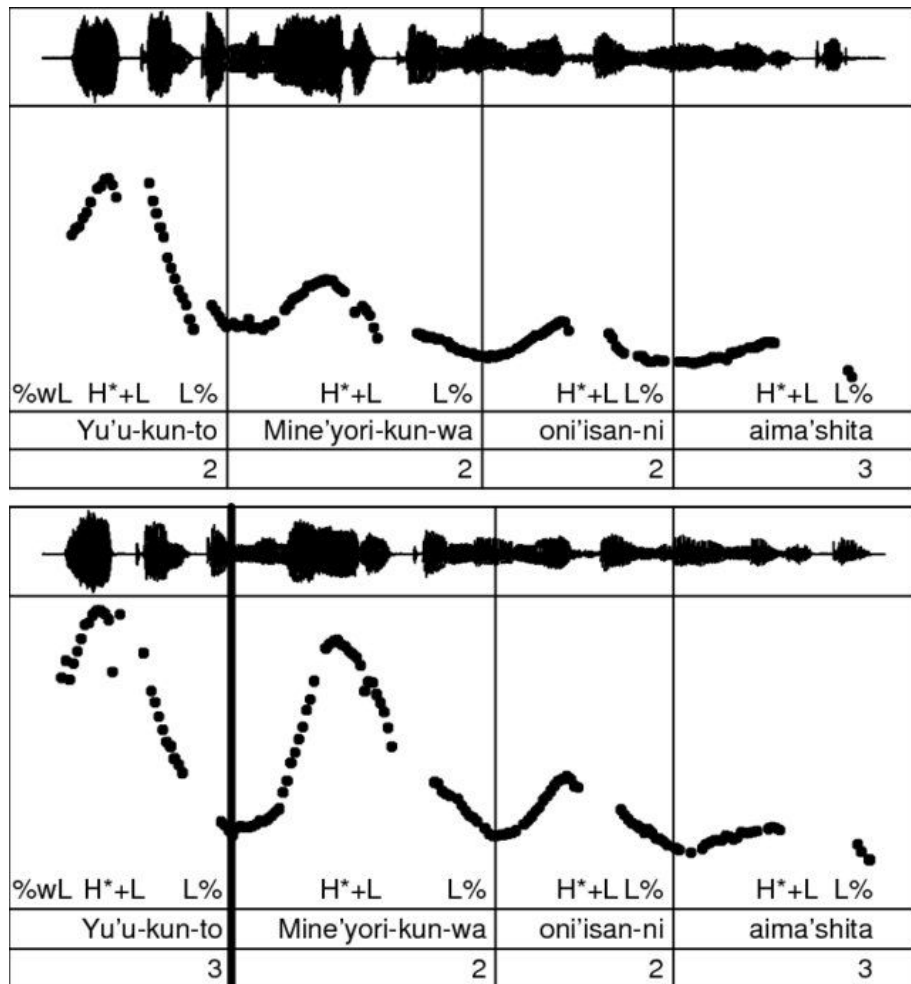
# Prosodic module : models


A model makes it possible to summarize all the intonation phenomena by a few parameters or intonation symbols.

TOBI Tonal Model, Stylization (Dutch school), Fujisaki Model: reports the acoustic characteristics of intonation


Ex: TOBI Tonal model:  $H^*+L$  a syllable which starts with a high accent and then decreases

Figure from Beckman, M. E., & Venditti, J. J. (2000). *Tagging prosody and discourse structure in elicited spontaneous speech.*

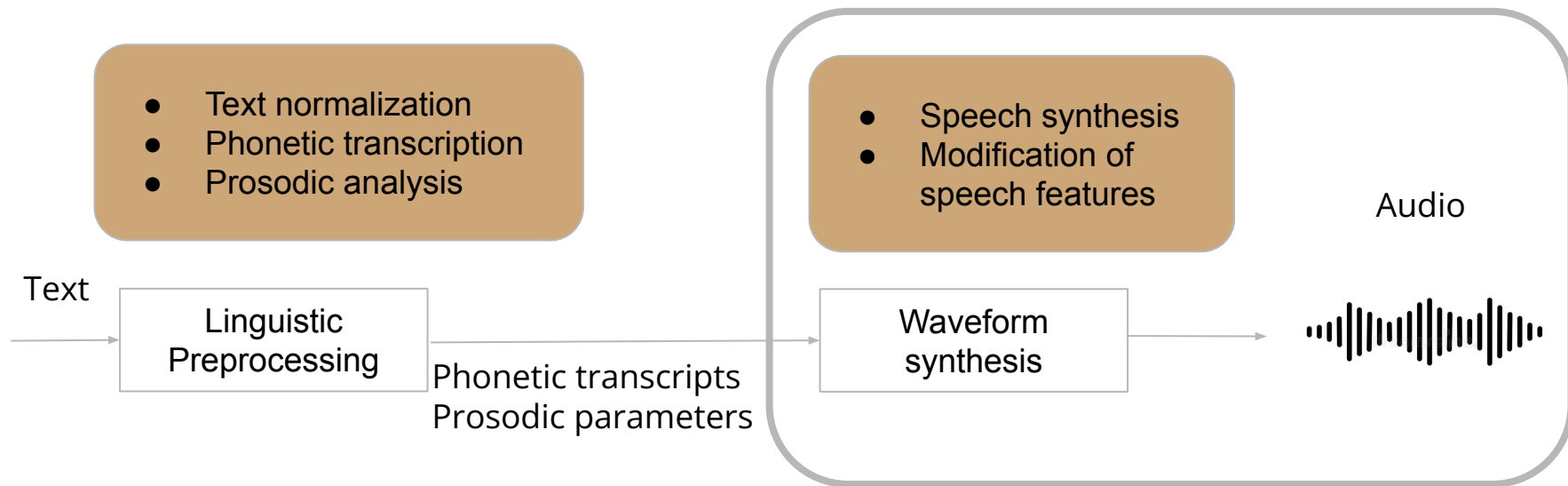




# Generation of speech signal/ Vocal synthesis



# Overall architecture of a two-block TTS





# Different approaches

1. Rule-Based Approaches [Klatt, 1980]
2. Synthesis by concatenation [Hunt & Black, 1996].
3. Parametric synthesis: generative synthesis from models (HMM) [Zen et al., 2009].
4. Deep learning approaches (WaveNet) [Van Den Oord et al., 2016].

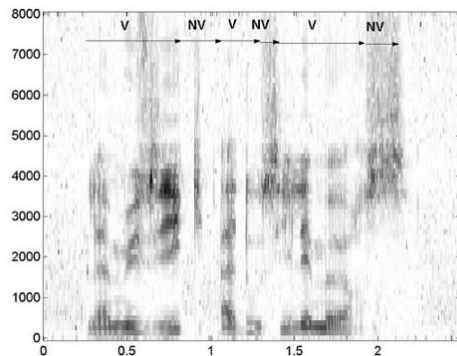
Examples :

<https://deepmind.com/blog/article/wavenet-generative-model-raw-audio>

# 1/ Rule-based approaches [Klatt, 1980]

**Objective:** to build the rules to create a sound signal from a sequence of given phonemes and the prosody (calculated from the linguistic analyses).

**Principle:** Reverse the process of reading the spectrogram, by doing formant synthesis



Spectrogram of the sentence : « la musique adoucit les mœurs »

# 1/ Rule-based approaches [Klatt, 1980]

## **Advantages:**

- Little data to store
- Integrates knowledge about speech

## **Disadvantages:**

- Long and tedious rule setting
- Rules depend largely on the language and to a lesser extent on the speaker

## 2/ Concatenative synthesis [Hunt & Black, 1996]

### Principle :

Assembling speech segments (stored in a database) corresponding to the phoneme sequence

Purely acoustic smoothing of discontinuities that may appear at the points of concatenation

Note : Requires only limited knowledge of the speech signal.

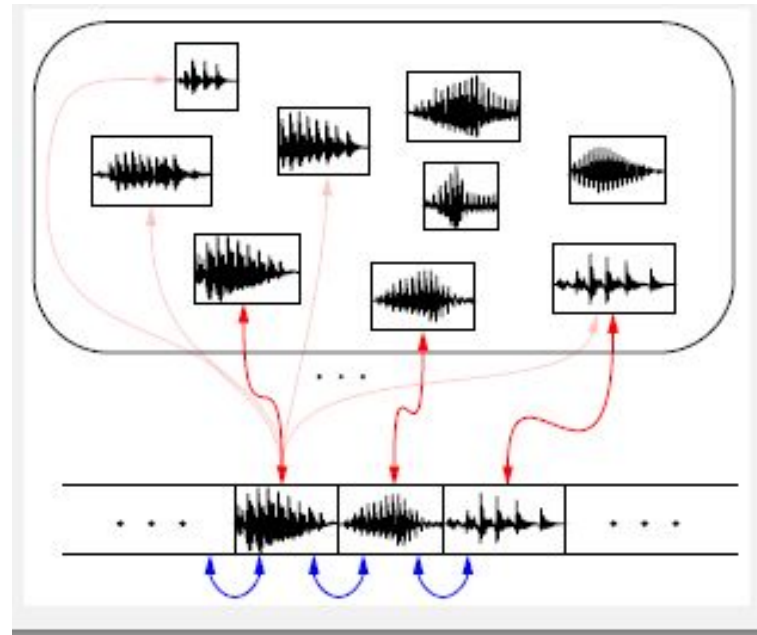


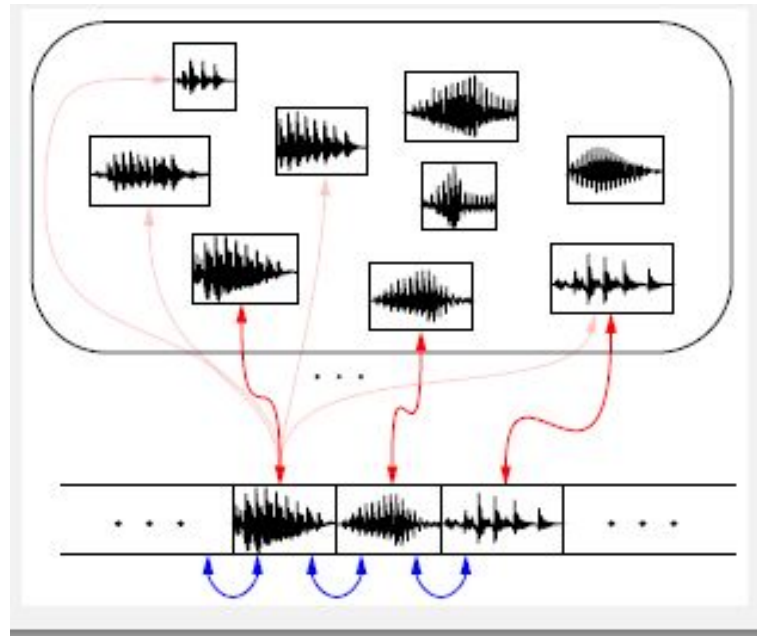
Schéma tiré de [1]

## 2/ Concatenative synthesis [Hunt & Black, 1996]

### Database of speech segments :

From mono-speaker speech recordings with a wide linguistic diversity

Split the acoustic signal in a relevant acoustic unit (the diphone)



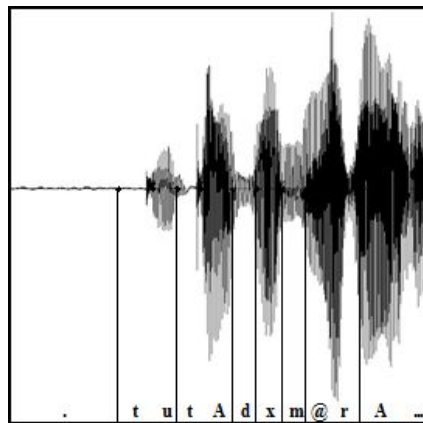
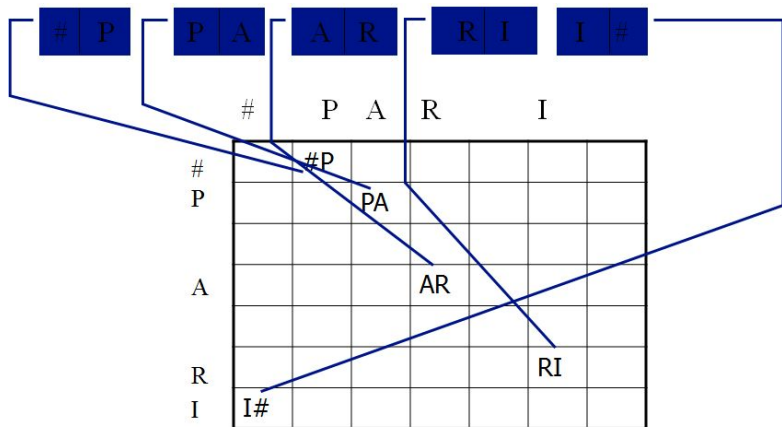
From [1]

## 2/ Concatenative synthesis

**Diphones:** (blue) acoustic unit that begins in the middle of the stable range of one phoneme and ends in the middle of the stable range of the next phoneme.

Allows for the inclusion of coarticulation phenomena.

Splitting signal into diphones requires to have an alignment between the signal and the phonemes



## 2/ Concatenative synthesis

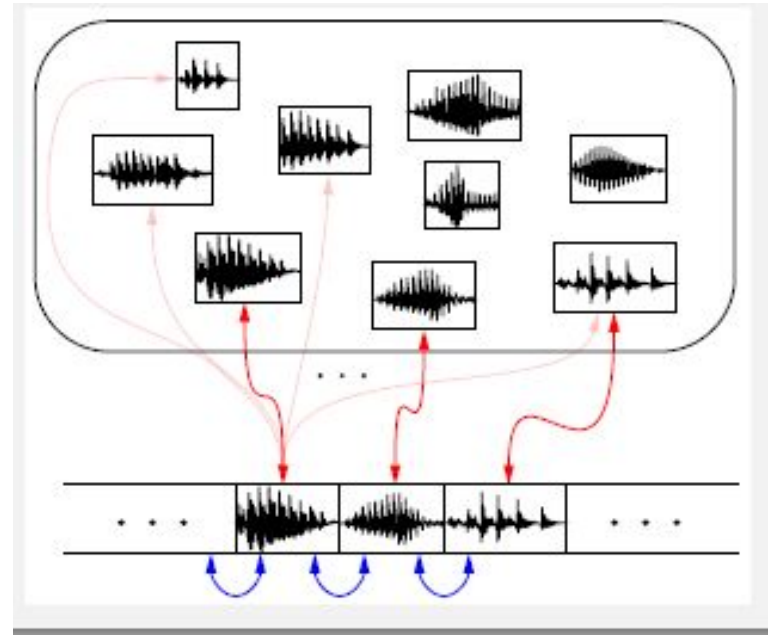
Considering the input phoneme sequence...

... select the synthesis units (speech segment) that will minimize future concatenation problems.

Two types of selection : static or dynamic

### **Static** selection

- Pair of phonemes of the input phonetic chain -> selection of the corresponding diphone signal
- Only one choice per unit



## 2/ Concatenative synthesis:

**Dynamic** selection among several instances of the same diphone

- Dataset: instances of the same diphone :
  - with different prosodies
  - positioned in different phonetic contexts.
- Choice made :
  - at the time of the synthesis
  - according to a global selection cost
- Selection cost depending on:
  - Representation cost:
    - phonetic context as close as possible to the phonetic string to be synthesized
    - prosody as close as possible to the prosody to be produced
  - Concatenation cost:
    - starts and ends with the fewest spectral discontinuities
- Method:
  - Use of dynamic programming (Viterbi)



# 3/Parametric synthesis: generative synthesis from models

Problem formulation of generative synthesis

Model-based, generative synthesis

$p(\text{speech} = \text{[waveform]} \mid \text{text} = \text{"Hello, my name is Heiga Zen."})$

Instead of text : use **phonetic and prosodic transcription**

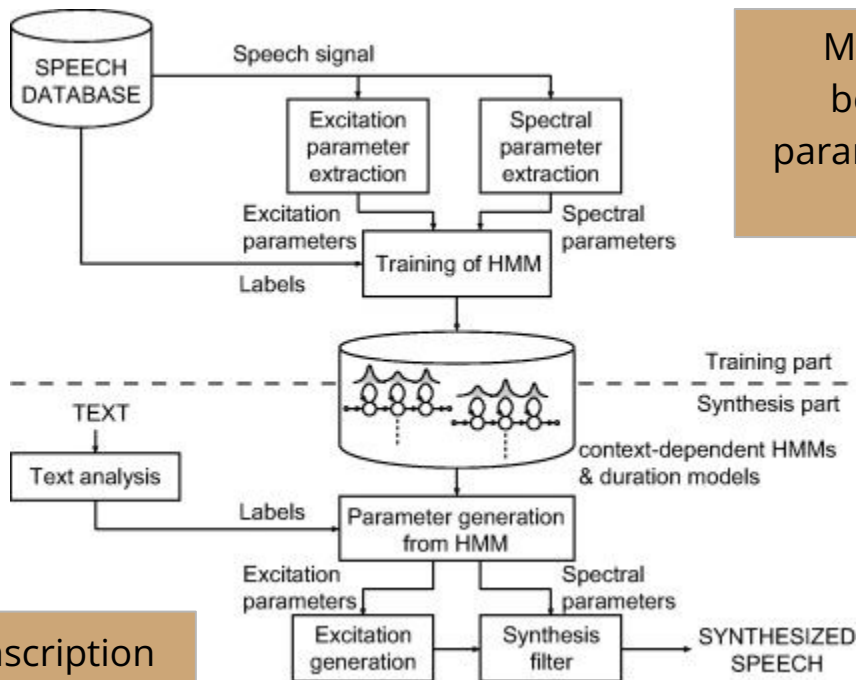
Instead of speech waveform : use **parameters of speech waveform** (e.g. source and filter parameters) and then use a **vocoder**

Model built from a training corpus :

Learn mapping between transcription  $\leftrightarrow$  parameters of speech waveform

# Ex : HMM synthesis

Corpus: speech recordings and their transcription (phonetic and prosodic transcription + emotional labels)



Models learned a mapping between transcription <-> parameters of speech waveform

phonetic and prosodic transcription  
With emotional and linguistic labels

Parameters of the  
speech form

Vocoder based on signal processing algorithm

# Wavenet vs. parametric speech synthesis

Replace the vocoder based on signal processing algorithms

By **directly modelling the raw waveform of the audio signal, one sample at a time.**

WaveNet predicts each audio sample compressed into 8-bits with a **256-way categorical classifier**

# WaveNet speech synthesis

WaveNet generates **raw audio waveforms** using a model (a fully convolutional neural network) which is:

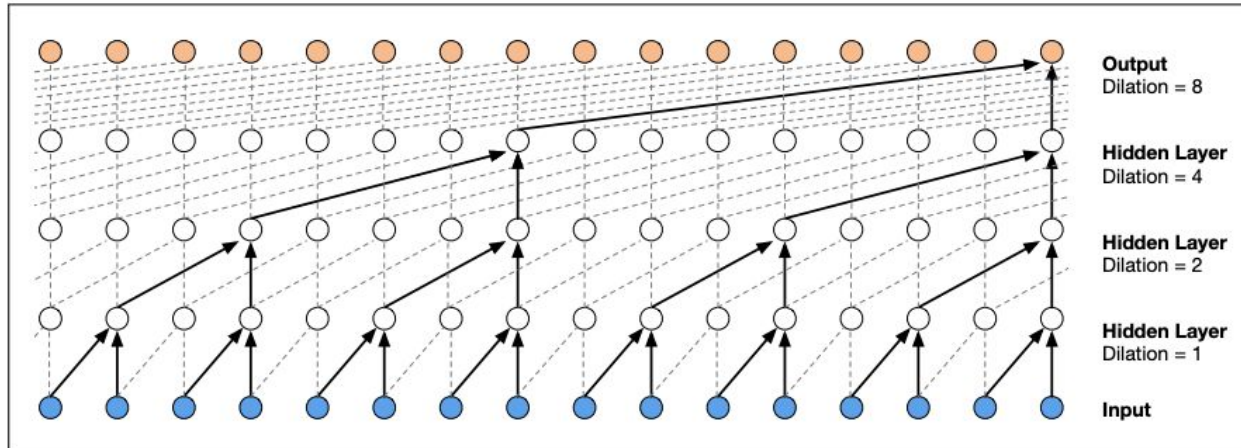
- Fully probabilistic (i.e., compute the probability for each audio sample)
- Autoregressive (i.e., the probability is conditioned on all previous audio samples)

$\mathbf{x} = \{x_0, x_1, \dots, x_{N-1}\}$  : raw waveform

$$p(\mathbf{x} \mid \lambda) = p(x_0, x_1, \dots, x_{N-1} \mid \lambda) = \prod_{n=0}^{N-1} p(x_n \mid x_0, \dots, x_{n-1}, \lambda)$$

# Wavenet speech synthesis

This probability distribution is modeled by a stack of **special convolution layers (causal dilated convolutions + a specific non-linearity function)**



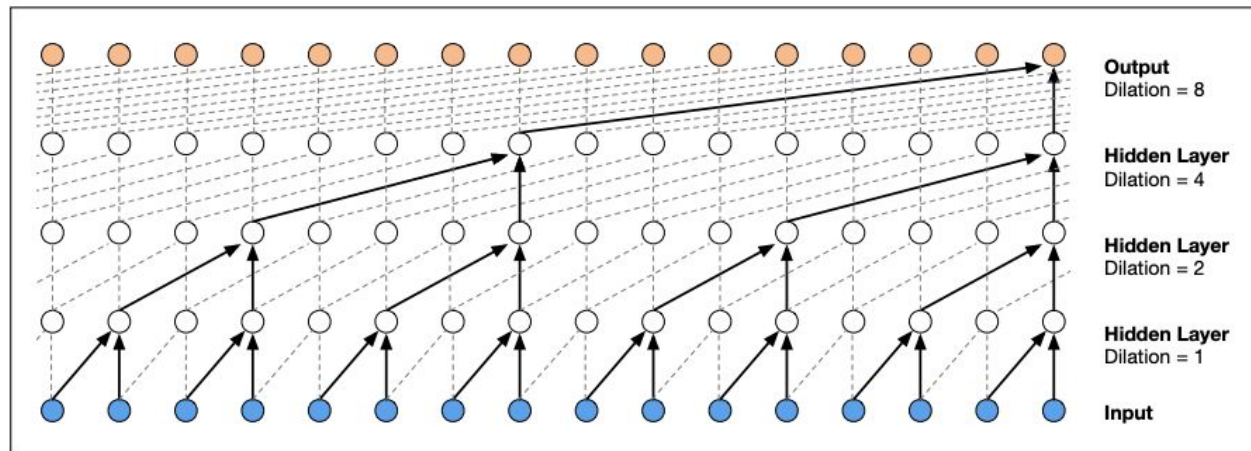
**Figure 26.15** Dilated convolutions, showing one dilation cycle size of 4, i.e., dilation values of 1, 2, 4, 8. Figure from [van den Oord et al. \(2016\)](#).

**Causal** convolutions look only at the past input (auto-regressive), rather than the future; the prediction of  $y_{t+1}$  can only depend on  $y_1, \dots, y_t$ , useful for autoregressive left-to-right processing.

**Dilated** convolutions : skipping input values  
Dilation = 4 : considers one out of every 4 values

# Wavenet speech synthesis

The output of the convolution layers is then passed through a softmax which makes 256-way decision (in order to obtain a sample compressed into 8 bits)



**Figure 26.15** Dilated convolutions, showing one dilation cycle size of 4, i.e., dilation values of 1, 2, 4, 8. Figure from [van den Oord et al. \(2016\)](#).

# Pre-requisite to use WaveNet as a TTS tool

transforming the text into a sequence of linguistic (including phonetic) features (which contain information about the current phoneme, syllable, word, etc.)

**Linguistic (and phonetic) features** : phone, syllable, word, phrase, and utterance-level features see exhaustive list here: [http://www.cs.columbia.edu/~ecooper/tts/lab\\_format.pdf](http://www.cs.columbia.edu/~ecooper/tts/lab_format.pdf))

- phone identities
- syllable stress
- the number of syllables in a word,
- position of the current syllable in a phrase,
- frame position of the phone
- phone duration (obtained using LSTM-RNN models trained to minimize Mean Squared Errors)
- F0 features (obtained using autoregressive CNN models trained to minimize Mean Squared Errors)

# Wavenet

What happened when trained without the text sequence?

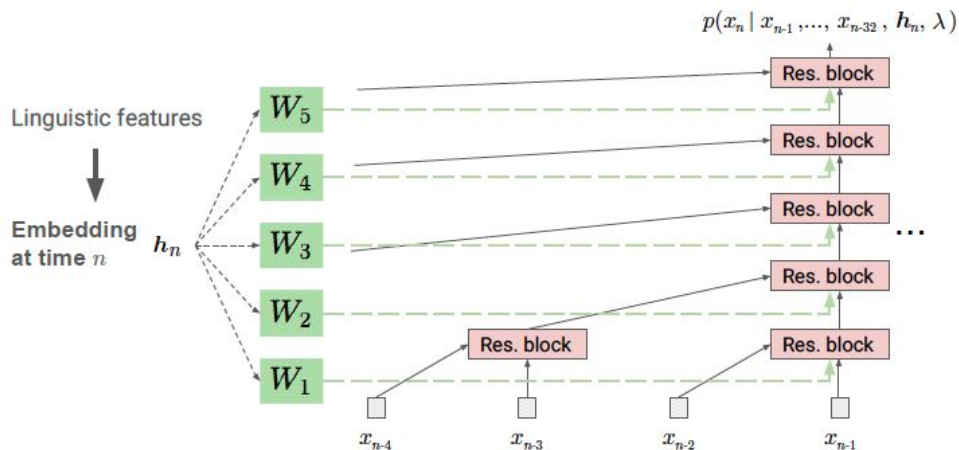
The network is able to generate speech with real words and word-like sounds (results in a kind of babbling)



# WaveNet speech synthesis: direct mapping from linguistics to waveform

Feeding this sequence of linguistic features (including phonetic features) into WaveNet

the probability is now conditioned on all previous audio samples **AND the sequence of linguistic features of the text** to synthesize





To go further



# Multimodal Laughter synthesis

Projet Ilhaire - Incorporating Laughter into Human Avatar Interactions:  
Research and Experiments

[https://www.youtube.com/watch?v=9MZwQdxlo1s&t=37s&ab\\_channel=NewScientist](https://www.youtube.com/watch?v=9MZwQdxlo1s&t=37s&ab_channel=NewScientist)



# References

[1] [Generative Model-Based Text-to-Speech Synthesis Andrew Senior \(DeepMind London\)](#)

<https://web.stanford.edu/~jurafsky/slp3/26.pdf> Section 26.6

T. Dutoit [27]

cours de F. Beaugendre [10]

D. Klatt. Real-time speech synthesis by rule. Journal of ASA, 68(S1):S18{S18, 1980.

A. Hunt and A. Black. Unit selection in a concatenative speech synthesis system using a large speech database. In Proc. ICASSP, pages 373{376, 1996.

# References

J. Benesty, M. Sondhi, Y. Huang, « Handbook of Speech Processing », Springer, 2008 (1176 pages)

C. d'Alessandro et G. Richard, "Synthèse de la parole à partir du texte", Collection Techniques de l'ingénieur, Paris, 2013

O. Boeffard et C. d'Alessandro, « Synthèse de la parole » dans Analyse, Synthèse et Codage de la parole, Hermès, Lavoisier, 2002.

R. Boite, H. Boulard, T. Dutoit, J. Hancq, and H. Leich. Traitement de la parole. Presses polytechniques et universitaires romandes, Lausanne, 2000.

H. Zen, K. Tokuda, A. Black « Statistical Parametric Speech Synthesis » , Speech Com. Volume 51, Issue 11, November 2009, Pages 1039–1064

Van Den Oord et al. "WaveNet : A generative model for raw audio.", 2016