



# Multimedia Data

## Speech Processing



# Contents

- Speech Signals
- Speech Output
- Speech Input



# Digital Speech Processing

- The handling of digitized speech
- Speech can be processed by humans or machines
  - Human speech
    - Based on spoken languages
    - Has *semantic* content
  - Machine speech (speech synthesis)
    - Computers translate message into speech

# Speech Signal

- Voiced speech signals
  - Have periodic structure over a certain time interval
  - Remain quasi-stationary for about 30ms or more
- The spectrum of some sounds
  - Involve up to five frequency maxima which are called *formants*
  - *Formants*: a characteristic component of the quality of an utterance



# Phonemes

- *Phonemes* are basic building blocks of language
- Phonetic studies have found out that 255 different phonemes are required to cover every spoken language in the world except the “clicking languages” used in Africa



# Speech Output

- Translation from an encoded description of a message into speech by a computer
- More correctly called **Text-To-Speech** conversion (**TTS**)
- Fair-quality text-to-speech software has been commercially available for various computers and workstations



# Speech Output

- The machine generation of speech
- The challenge
  - How to generate these signals in real time
  - How to convert text to speech automatically
  - ⇒ Use limited vocabulary
- Speech output techniques
  1. Reproducible speech play out
  2. Sound concatenation in the time domain
  3. Sound concatenation in the frequency domain

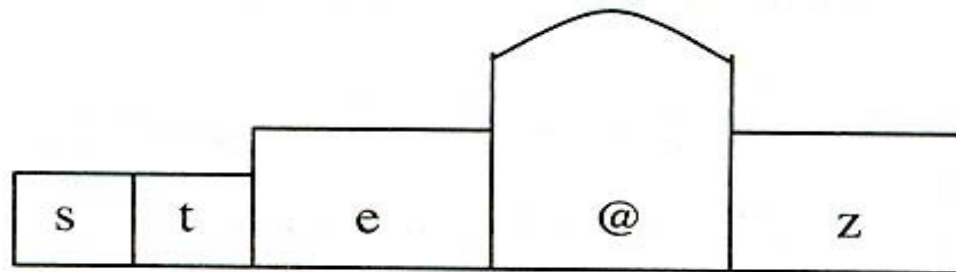


## Reproducible Speech Play out

- A straightforward method
- The speech is spoken by a human and recorded
- The stored sequence is played out

# Sound Concatenation in the Time Domain

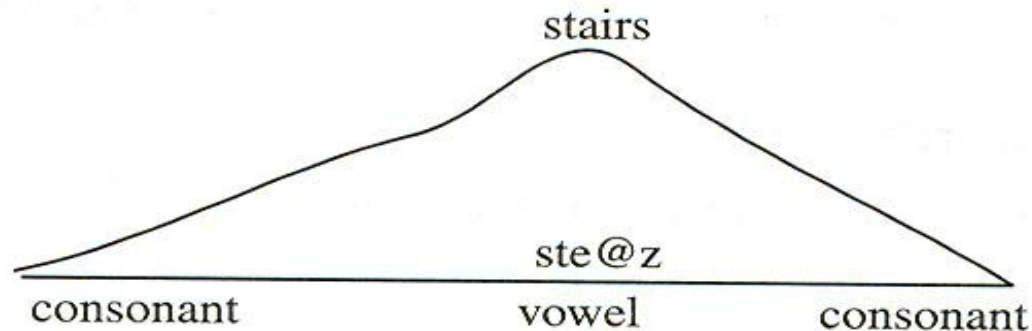
- A composition on various levels
  - Sound concatenation of a phoneme in the time range



“stairs”

# Sound concatenation of a word in Time Domain (b)

- The problem of phoneme concatenation
  - Transitions between the phonemes
  - Solution: storing the entire word



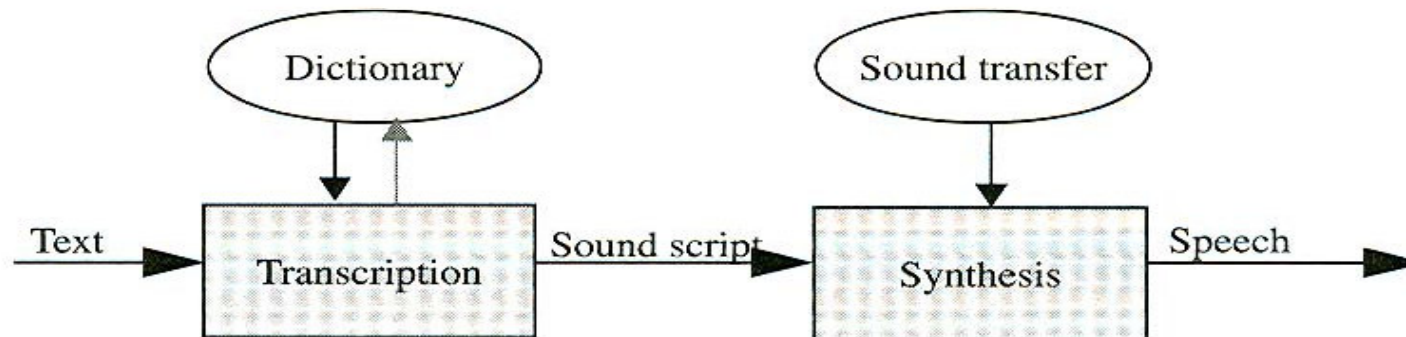


# Sound Concatenation in the Frequency Domain

- By formant synthesis
- Formant synthesizers
  - Use a model of the *vocal tract* and a set of rules to generate speech
  - The model uses frequencies, energies, pitch, and other acoustic-phonetic parameters as control variables
  - These systems can achieve high intelligibility
  - The problem is these systems do not sound very natural since it is very difficult to describe accurately the process of speech generation in a set of rules

# TTS

- Used to transform an existing text into an acoustic signal





# Components

- Transcription step (TTP)
  - Translation of the text into the corresponding phonetic transcription
  - Having a software solution
- Synthesis step
  - Conversion of the phonetic transcription into an acoustic speech signal, where concatenation can be in the time or frequency range
  - Involving signal processors or dedicated processors

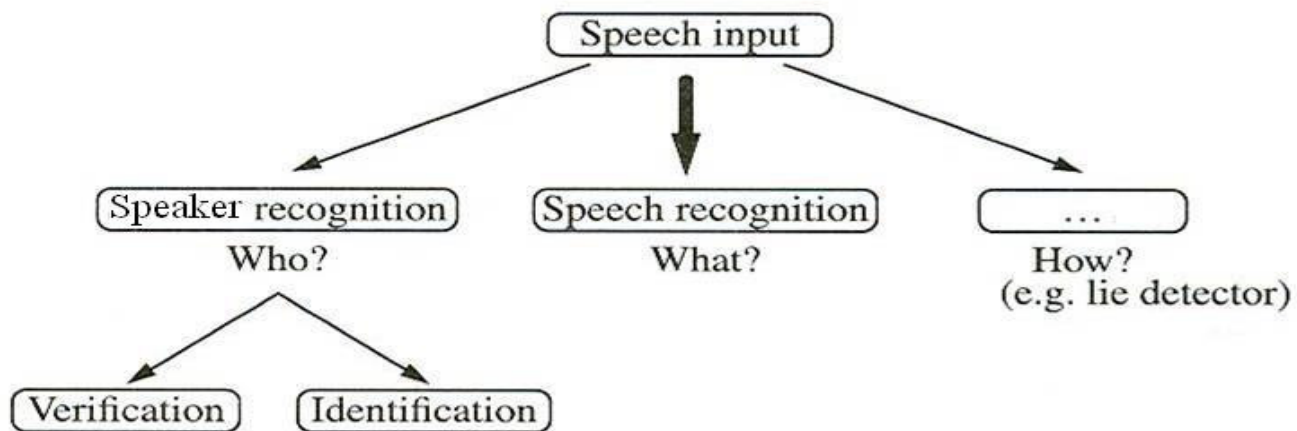


# Speech Synthesis Systems

- AT&T Watson Speech Synthesis
- BeSTspeech from Berkeley Speech Technologies, Inc. (BST)
- Creative TextOLE
- DECtalk:Text-to-Speech
- Windows SDK
- Lernout and Hauspie Text-to-Speech Windows SDK
- MBROLA: Free Speech Synthesis Project
- Tinytalk

# Speech Input

- Speech Input Application





# Speech Input Context

- Who?

- Human speech has certain speaker-dependent characteristics

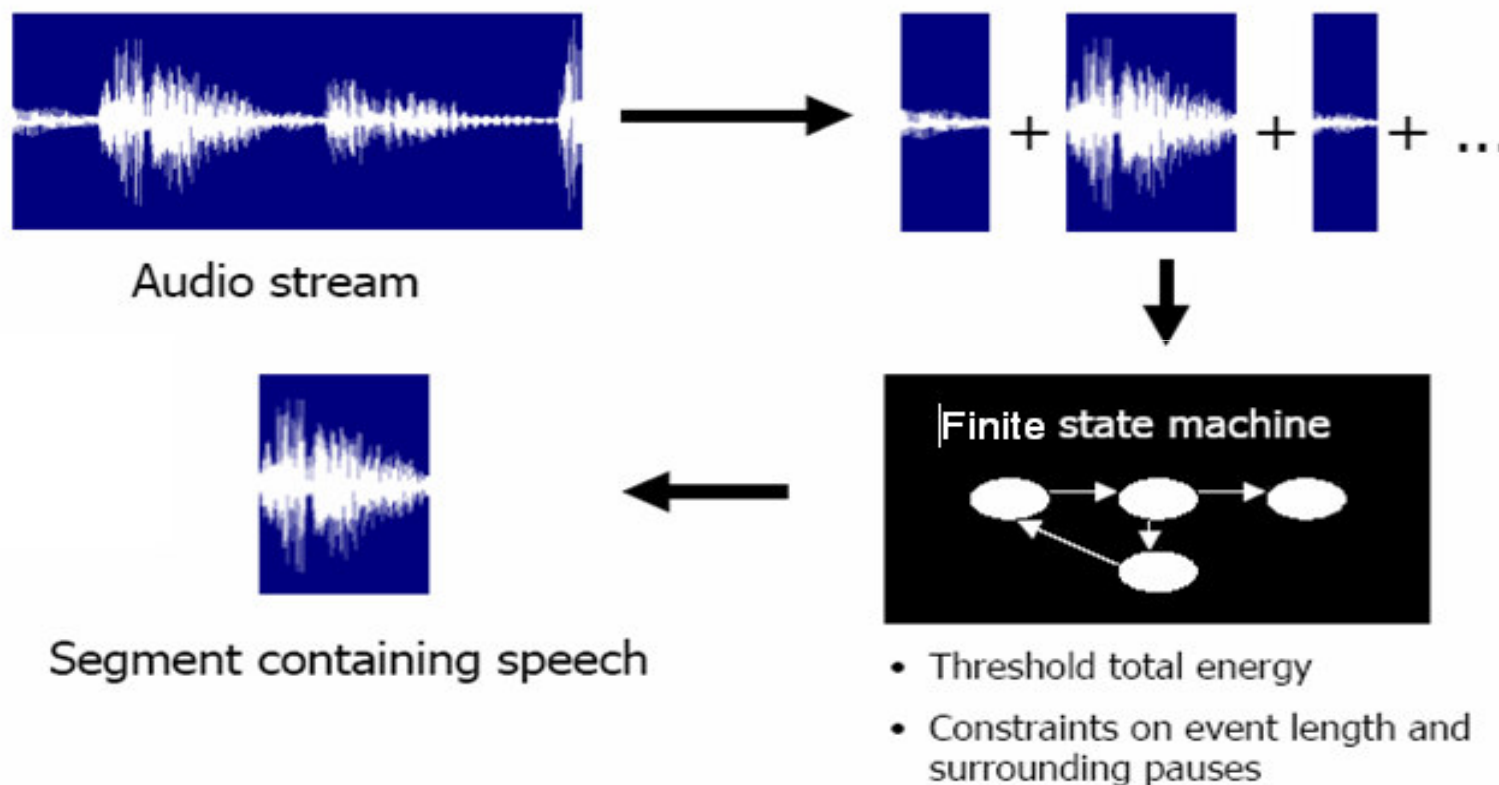
- What?

- Detecting the speech contents themselves

- How?

- How a speech sample should be studied
  - lie detector

# Who Spoke this stuff



Finite State Machines are the key to many temporal data analysis problems

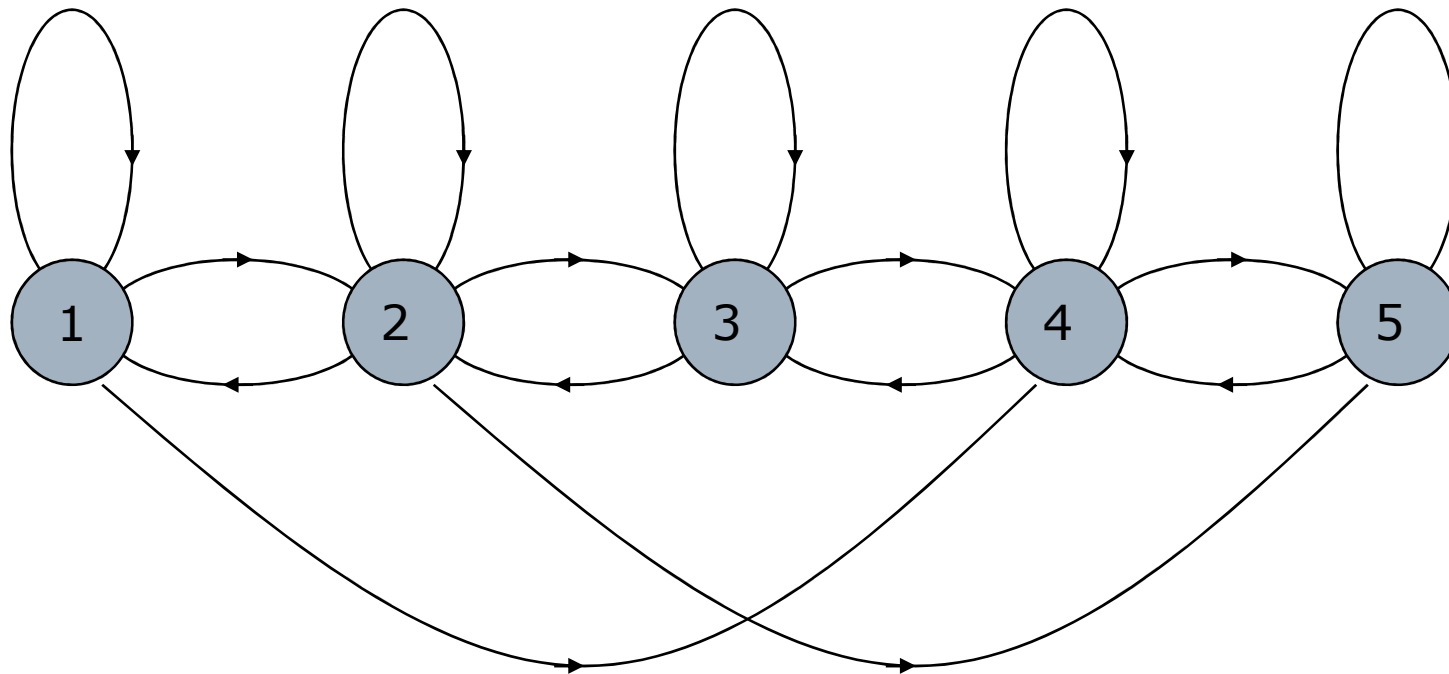


# Temporal Data and FSM

- **Finite State Machines (FSM)** are computational machines with finite number of states and a transition between each of the states
- Transitions between states is caused because of an input in a state
- Any temporal sequence could be modeled as a series of states in an FSM
- Used for modeling different speech units (phonemes, syllables, words, ...) in recognition

# Example of modeling Temporal Sequence in FSM

Phonemes in a word



**In this case the event that causes a transition is a change in voice due to changes in phonemes**



# Person Recognition Through Speech

- We can find a template of how people generally speak phonemes
- For every person we could find how he/she varies from the general way of speaking the phonemes
- This difference between average phoneme and person specific phoneme could be studied over time using FSM
- A given sequence in an FSM recognizes the person... just the same way in written word recognition, the sequence of “i” followed by “d” followed by “i” followed by “o” followed by “m” means “idiom”



# Speech Recognition

- Normally achieved by drawing various comparisons
  - Problems: Factors affecting recognition quality:
    - *continuous speech*
    - *speaker variability*
    - *vocabulary size*
    - *environmental noise*
    - *dialects*
    - *emotional pronunciations*
    - *state of a speaker*
- etc.



# System Development

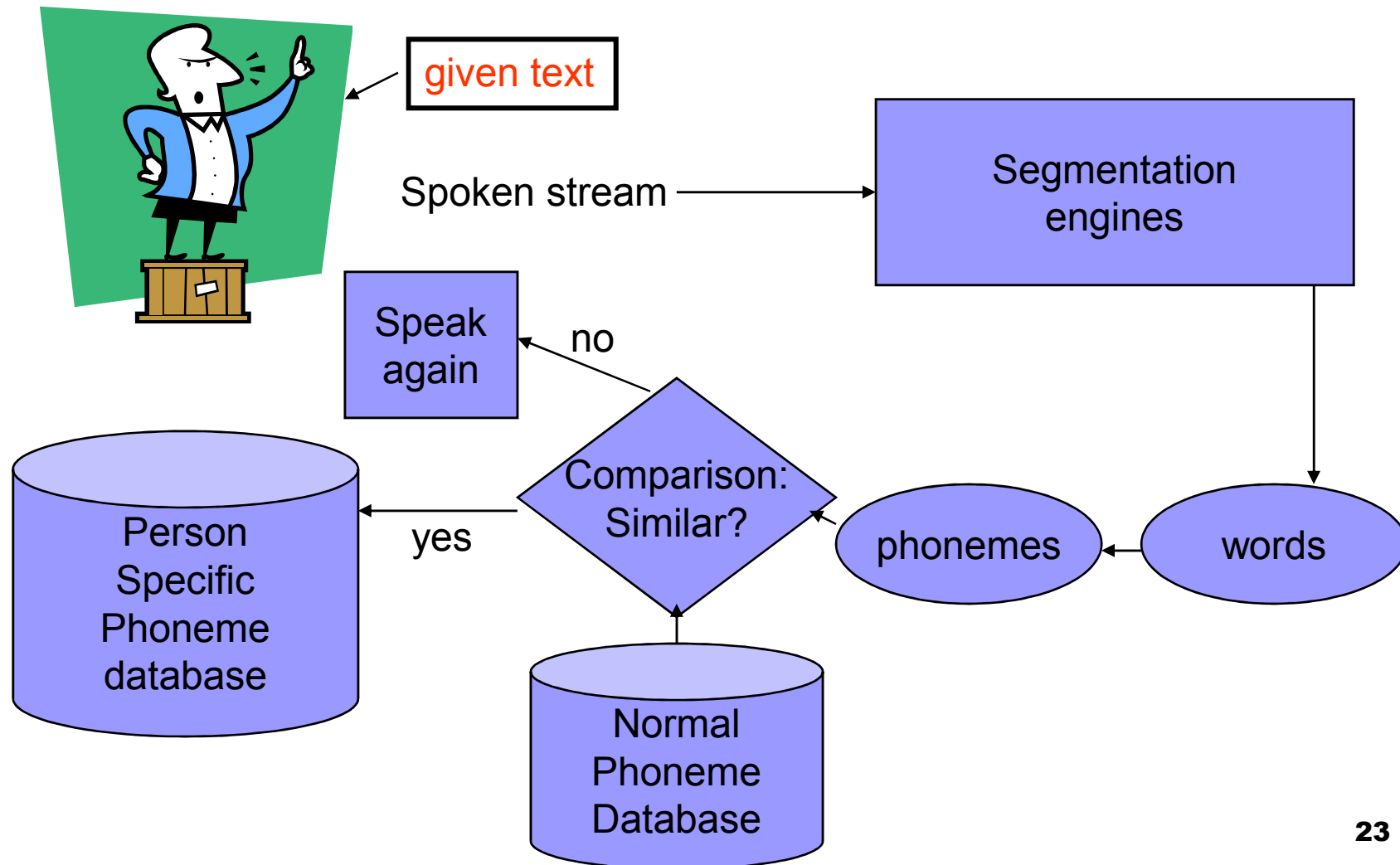
- Training

- Capturing the phonemes

- Training the Finite State Machine

- Testing

# Speech Recognition (One possible Scenario)





## Training the FSM

- In general all languages including English have *syntax* and *semantics*
- There are methods to encode syntax and semantics into the FSM
- Consider for example what is the probability in English that spoken “a” follows spoken “t” as compared to spoken “x” followed by spoken “z”



## Training-II

- We can take a big library of spoken utterances of words, and use that to train a FSM
- The idea is to find the *probability* of transitioning from one phoneme to another
- By encoding the probability in an FSM, recognition becomes much easier



# How to encode transition probabilities

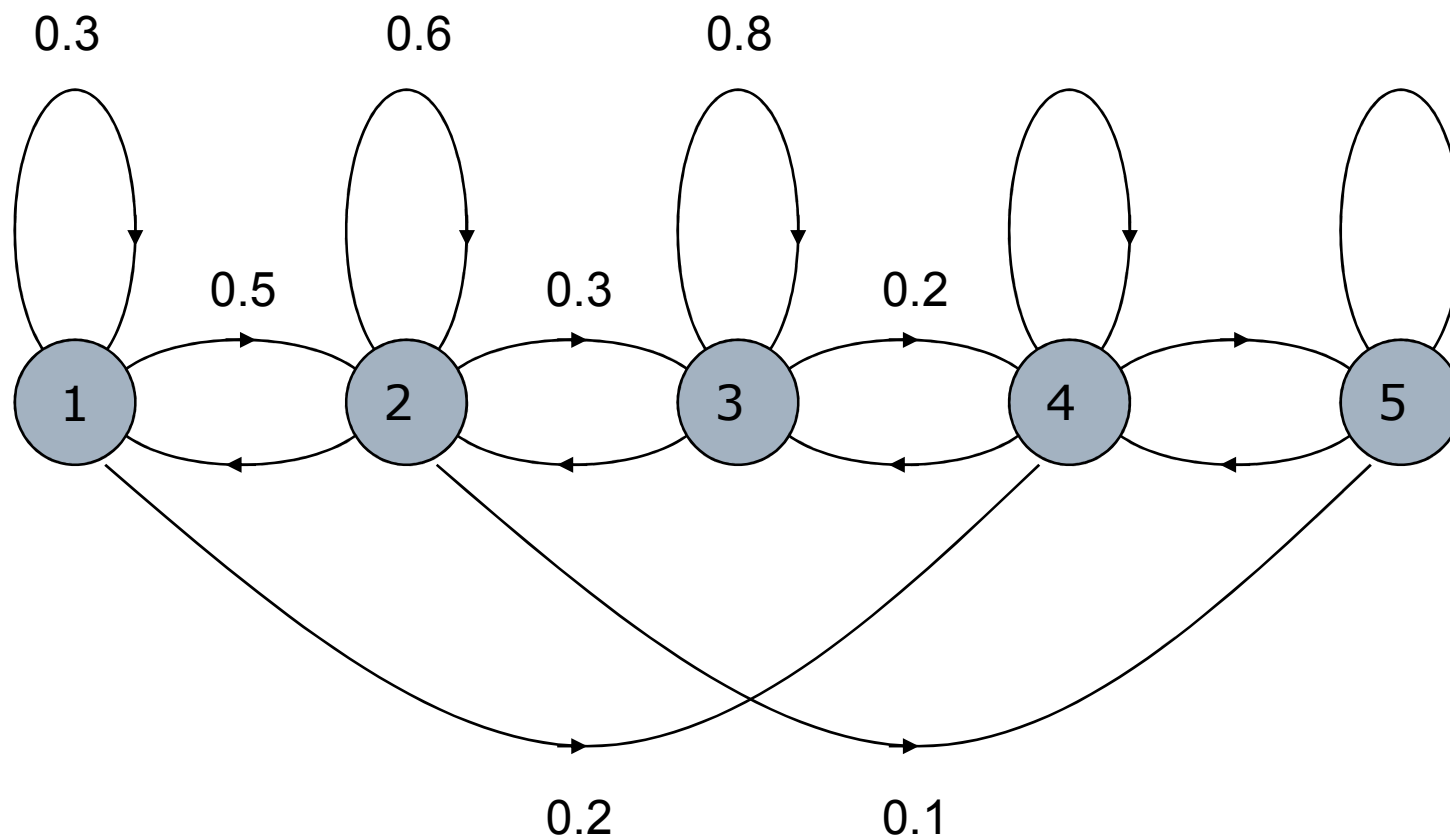
- Consider the transition between spoken “*a*” and spoken “*t*”
- Probability of the transition from *a* to *t* is number of times *a* is spoken and followed by *t* divided by total number of times *a* is spoken
- This is done for all  $P$  phonemes of the language and for each phoneme we have  $P$  transitions to find



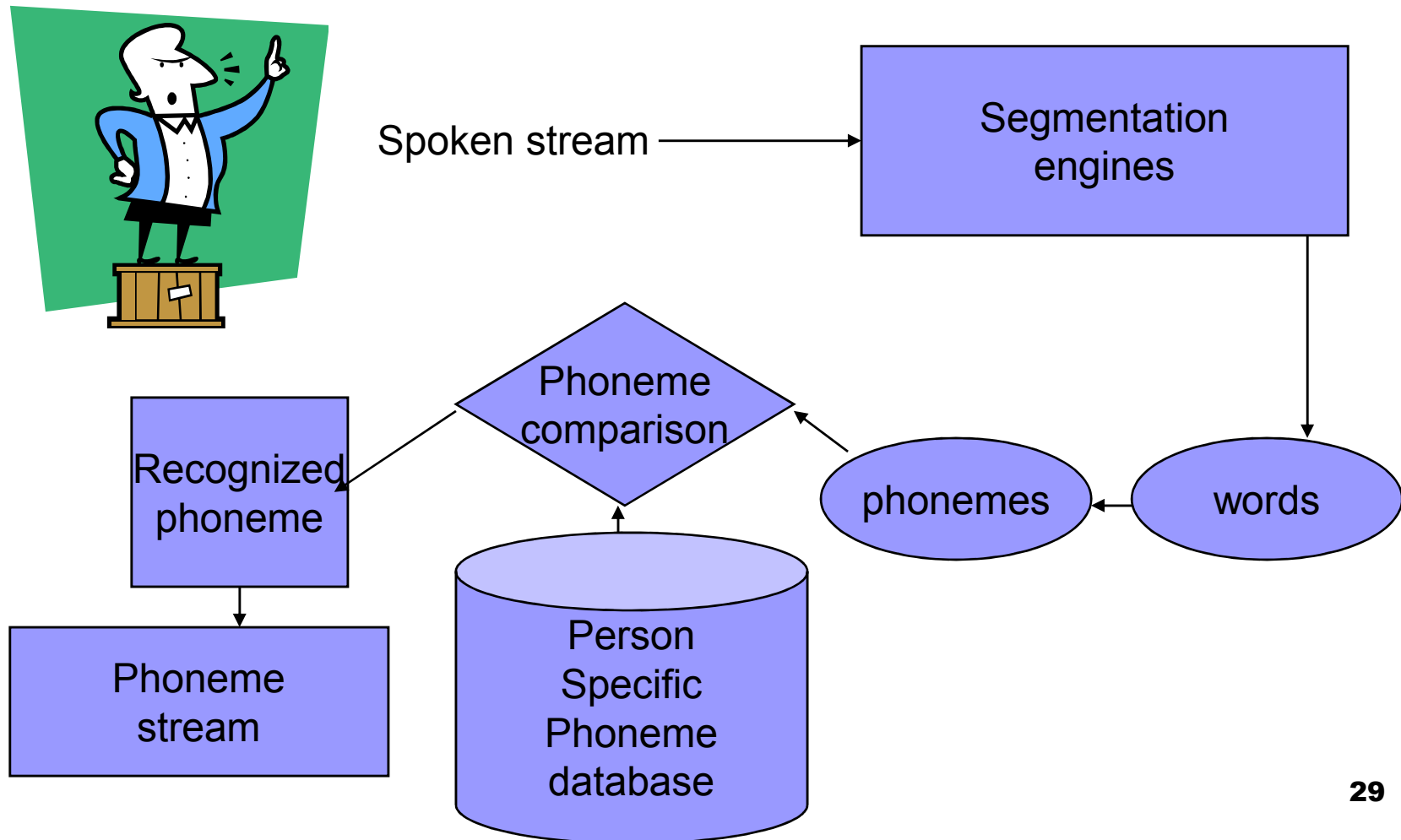
# The actual algorithm

- The actual algorithm is more time efficient
- The approach used is called *Hidden Markov Model* (HMM)
- It uses the probability theory of Markov chain and the basic algorithm is similar to one highlighted in previous slide
- Markovian property means the present state is influenced by the past states
- One of the most used algorithms in temporal modeling and sequence modeling

# Markov Model Training



# Testing

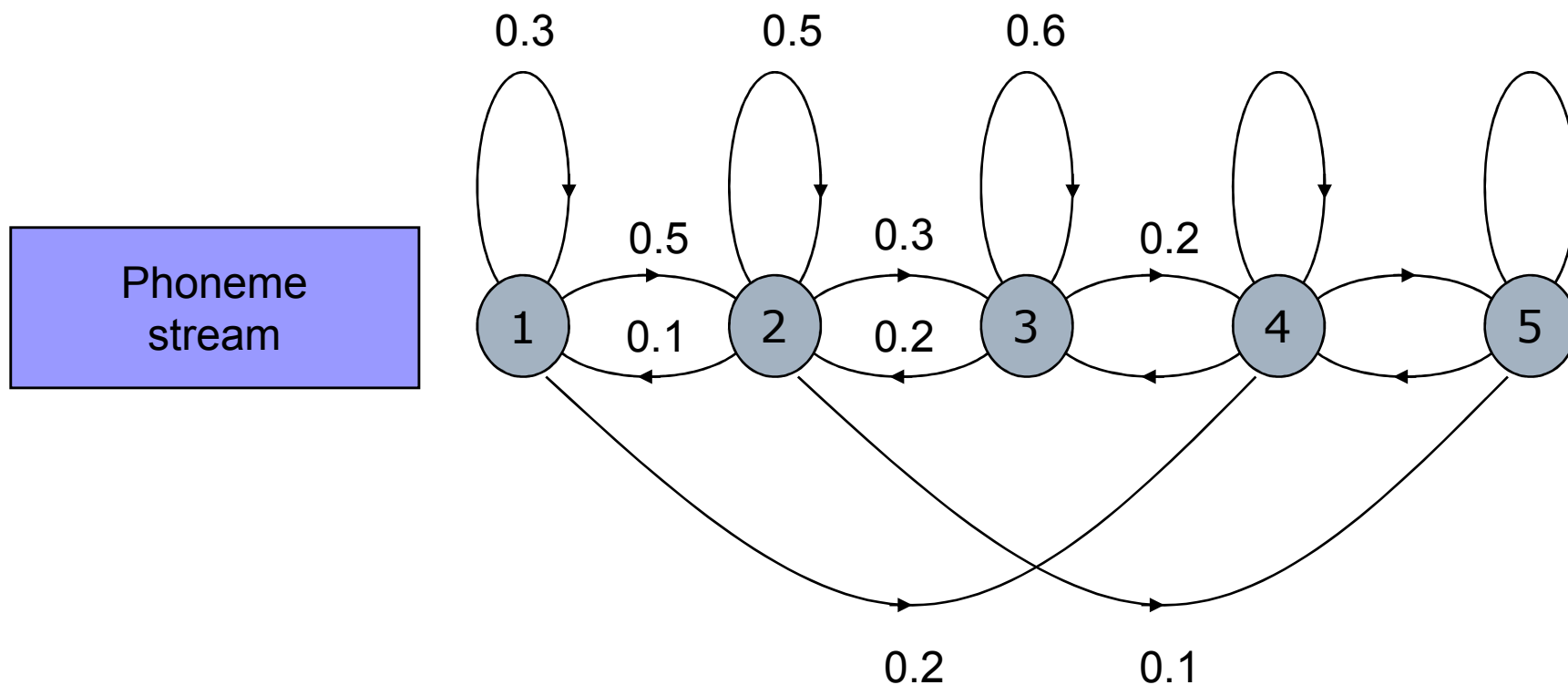




# The Phoneme Comparison

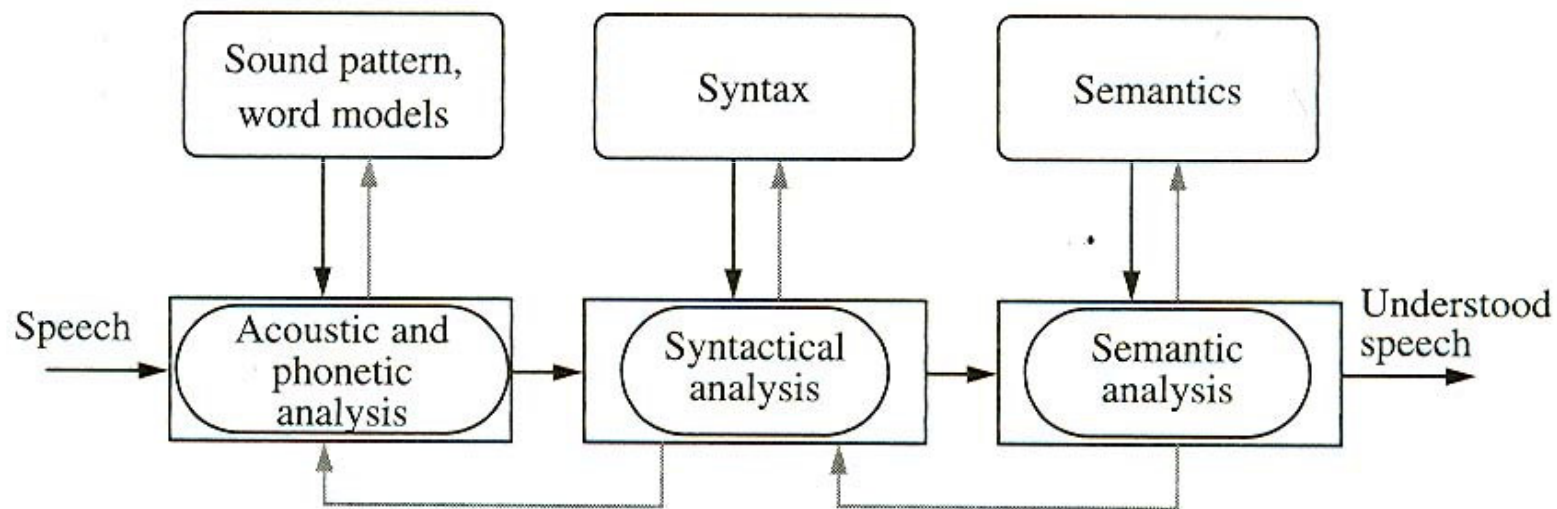
- The phoneme comparison engines are not very good...
- That is why we need to pass the stream through the FSM

# Testing II



**Go through all the transitions Possible for all possible start elements..  
Wherever the probability as multiplied is maximum is the recognized word.  
Real algorithms use tricks to lower the computational complexity**

# Higher-level Processes in Speech Recognition





# Speech Recognition Components

1. Acoustic and phonetic analysis
  - Referring to the characteristic properties of the chosen method (in the time or frequency range)
2. Syntactic analysis
  - Detect errors in the first run
3. Semantic analysis
  - Analyze the semantics of the speech sequence recognized
  - Detect errors for the previous decision process and remove them



# Speaker-Dependent Speech Input Systems

- Developed to operate for a single speaker
- Recognition of a speaker based on his or her voice
- Easier to develop, cheaper, and more accurate
- Not flexible as speaker-adaptive or speaker-independent systems
- The cost of “training”
  - The speaker is normally requested to read specific speech sequences



# Speaker-Independent Speech Input Systems

- Developed to operate for any speaker of a particular type (for example, American English)
- More **difficult** to develop, more expensive
- **Accuracy** is lower than speaker-dependent systems
- More **flexible**



# Speaker-Adaptive Speech Input Systems

- Developed to adapt its operation to the characteristics of new speakers
- Between Speaker-Independent and Speaker-Dependent Systems



# Speech Recognition Systems

- AT&T Watson Speech Recognition
- Cambridge Voice for Windows
- Dragon Dictate for Windows
- Dragon Dictation Products (*Naturally Speaking*)
- IBM Viavoice Dictation
- Kurzweil Speech Recognition
- Lernout & Hauspie ASR SDK
- Listen for Windows 2.0 from Verbex Voice Systems
- Microsoft Speech Recognition
- Philips Speech Recognition
- VoiceAssist for Windows from Creative Labs, Inc.
- Whisper



# Reference

- <http://cubicstore.eas.asu.edu/kanav/mm>