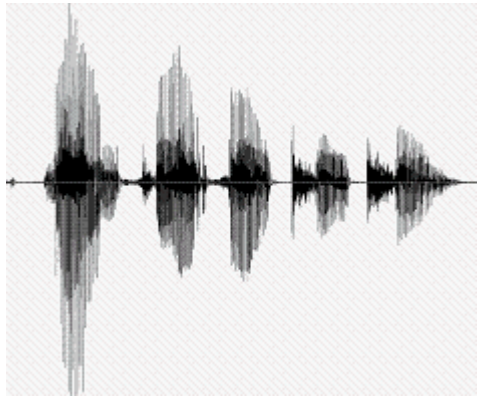
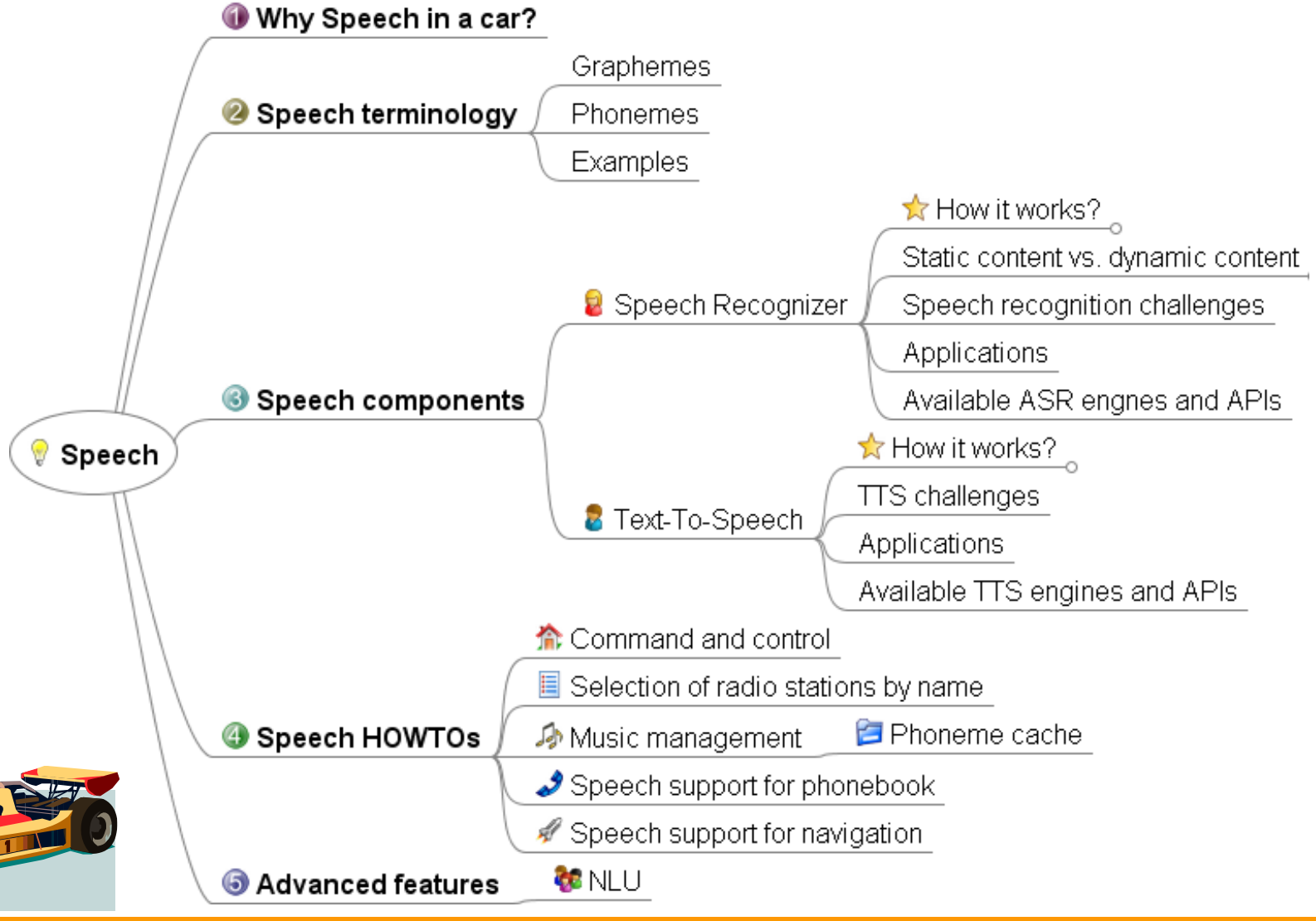
Curs 08

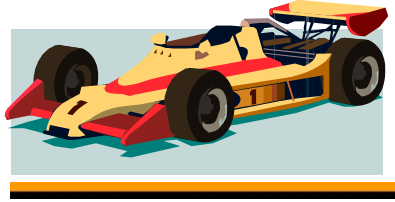
Speech

Cristian Miron



Summary


Speech
1 Why Speech in a car?
2 Speech terminology
Graphemes
Phonemes
Examples
3 Speech components
Speech Recognizer
How it works?
Static content vs. dynamic content
Speech recognition challenges
Applications
Available ASR engnes and APIs
Text-To-Speech
How it works?
TTS challenges
Applications
Available TTS engines and APIs
4 Speech HOWTOs
Command and control
Selection of radio stations by name
Music management
Phoneme cache
Speech support for phonebook
Speech support for navigation
5 Advanced features
NLU



Why Speech?

- Difficulties on using touch screen or the buttons from board
- Using Speech, the drivers' eyes are always on the road (give commands to machine, hear announcements from navigation, etc.)
- What can be done with speech in a car:
  - Control media player / radio stations
  - Control navigation
  - Control phone book and phone activities
  - Find POI locations (POI : point of interests)
  - Get machine status
  - E-mail/SMS reading
  - Customizations: language, voice

Speech terminology

- **Grapheme** is a fundamental unit in a written language:
  - Alphabetic letters: Latin, Greek, Cyrillic etc. letters

    Я
  - Chinese characters

  - Numerical digits

    2
  - Punctuation marks

    ! ? :

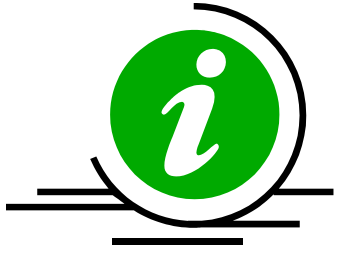
Speech terminology

- **Phoneme:** the smallest unit of speech that affects the meaning of a word. A sound unit.
  - The **c** in **c**at and the **m** in **m**at are phonemes.

- Examples in English:

/sʌm/ *sum*	[bɛt] *bet*
/sʌn/ *sun*	[bɛd] *bed*
/sʌŋ/ *sung*	

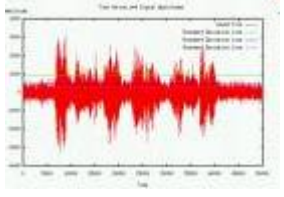
Speech recognition



- What is speech recognition? How it works?
- Static content versus dynamic content
- Challenges
- Applications
- Available ASR engines and APIs

What is speech recognition?

- **Speech recognition** converts spoken words to machine-readable input



algorithm

"#'spit&S#"

"speech"

Most common speech recognition algorithms and characteristics

- Algorithms
  - Hidden Markov model (HMM)-based speech recognition (widely used in many systems)
  - Dynamic time warping (DTW)-based speech recognition

- The performance of speech recognition systems is usually specified in terms of *accuracy and speed*
  - **Accuracy** may be measured in terms of performance accuracy which is usually rated with word error rate (WER)
  - **Speed** is measured with the real time factor (If it takes time *P* to process an input of duration *I*, the real time factor is defined as RTF = P / I)

- *Optimal conditions* usually assume that users:
  - have speech characteristics which match the training data,
  - can achieve proper speaker adaptation, and
  - work in a clean noise environment (e.g. quiet office or laboratory space).

g2p module

- G2P ("grapheme to phoneme" module) allows creation of the phonemes (transcriptions) from a grapheme. A grapheme could have one or more than one transcriptions.

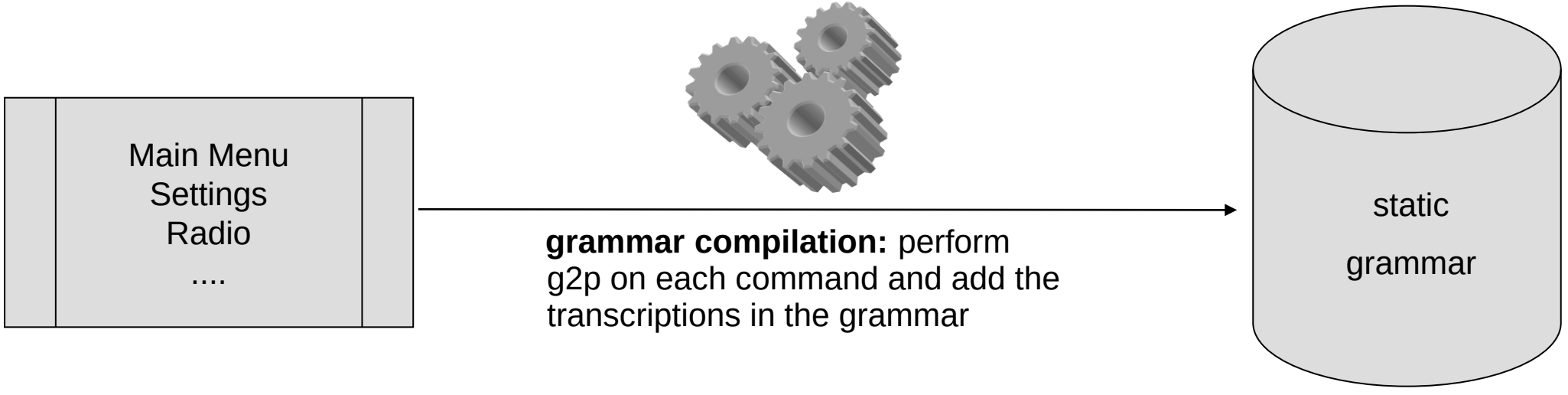
- G2P module is language depended, a word could be pronounced differently in different languages:
  - Example:
    - Radio in English is pronounced "#'R+e&I.di.o&U#"
    - Radio in German is pronounced "#'Ra:.di:.o:#"

Grammars

- A grammar is a collection of phonemes (transcriptions) and the associated graphemes
- The content of the grammars can be static or dynamic

- **Static content**
  - This type of grammar is used for "Command & control" features, when the words to be recognized are the same for all users
  - Examples of commands: "Main menu", "Settings", "Radio", "FM"
  - The developer has to use a Recognition engine tool to compile the grammar: which will use as an input a list with commands and as output there will be a binary file containing the transcriptions for all commands.
  - The recognition process will be like this: the user will say a commands, the recognizer will apply an algorithm like HMM and as output it will generate a list with best matches for that audio signal and a confidence level for each command recognized
  - Most of the time, for command and control, only the first command from the list with possible commands will be used (the one with the highest confidence level)

Static grammars



Grammars

- **Dynamic grammars**
  - For features like music management or selection of radio stations by name, the words which need to be recognized are not the same for all users (it depends of the titles of songs listen by each user or by the radio stations present on his area).
  - The solution is to use text enrollment functionality, which allows the application to add at run-time new words in the recognizer; this means that the g2p action will be performed also on the fly.

Radio stations

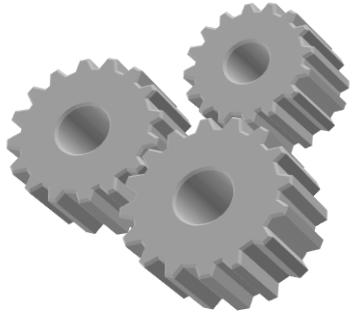
Get radio station names

from **radio tuner**

Kiss FM
Radio 21
INFO PRO
....

Perform g2p for radio station names for current language

#'kIs_'2Ef.'Em#
#'R+e&I.di.o&U_'2twEn.ti.'w^n#
#'In.fo&U_'pR+o&U#
....



add to slot

station <STATION_NAME>

dynamic

grammar

Music management

Use **id3 parser** to read from mp3 files titles, artists, composers, genre, album. etc.

Title: One
Artist: U2,
Album: Achtung Baby,
Genre: rock
...

Perform g2p for all titles, artists, etc.

transcriptions

add to slot:

- *title <DYN_TITLE>*
- *artist <DYN_ARTIST>*

dynamic grammar

Phonebook

Download contacts from phone via **bluetooth**

Madalina Ionescu, 0725xxxxxx
Irinel Ion, 0743xxxxxx

Perform g2p for all names

transcriptions

add to slot:

- *title <DYN_NAME>*

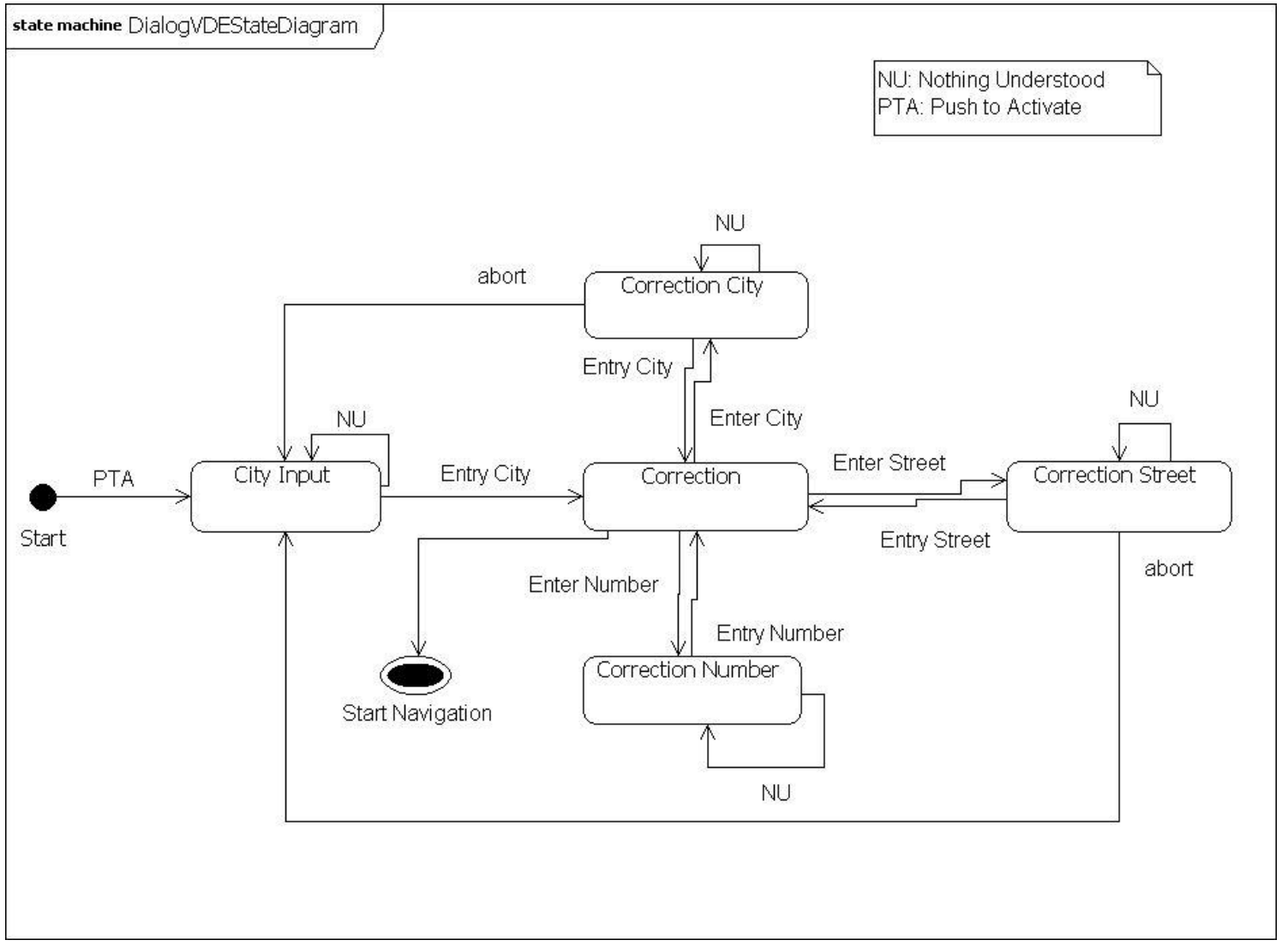
dynamic grammar

Naviation

- Voice destination entry:
  - **App:** Please say the name of the city!
  - **User:** London
  - **App:** City name: London. Please say the name of the street!
  - **User:** Main Street
  - **App:** Street name: Main. Please say the street number!
  - **User:** 56
  - **App:** Street number: 56. Start navigation?

- One short destination
  - **User:** London, 56th Main Street

Example of SUI (Speech User Interface) state diagram: Navigation



Applications

- Health care
- Military
  - High-performance fighter aircraft
  - Helicopters
  - Battle management
  - Training air traffic controllers
- Telephony and other domains
- People with disabilities
- Mobile telephony, including mobile email
- Robotics
- Video games
- Home automation
- Automotive speech recognition
- Hands-free computing: voice command recognition computer user interface

Speech Recognition engines

Microsoft Speech Server

IBM: WebSphere Voice Server

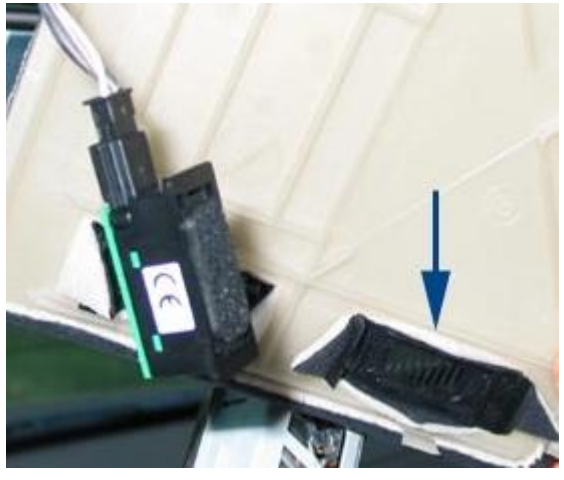
Nuance: VoCon

VoiceBox Speech Recognition

Challenges of speech recognition in automotive industry

- Noisy environment, the noise level of the signal is between 20dB and 5db

- Embedded platforms with restrictions related to CPU and memory/space storage



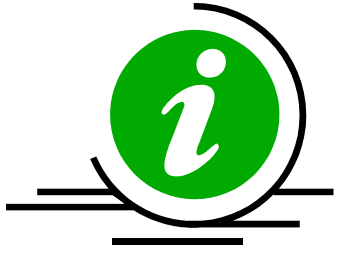
- Cheap microphones (low cost) placed 30-100 cm from speaker

Speech recognition in automotive industry

- Ford SYNC (developed by Ford and Microsoft)

- Continental MMP (developed by Continental/Siemens)

Speech synthesis



- What is Speech synthesis? How it works?
- Challenges
- Applications
- Available TTS engines and APIs

What is speech synthesis?

- Speech synthesis is the artificial production of human speech.
- Can be implemented in software or hardware
- A **text-to-speech (TTS)** system converts normal language text into speech
- Synthesized speech can be created by concatenating pieces of recorded speech that are stored in a database

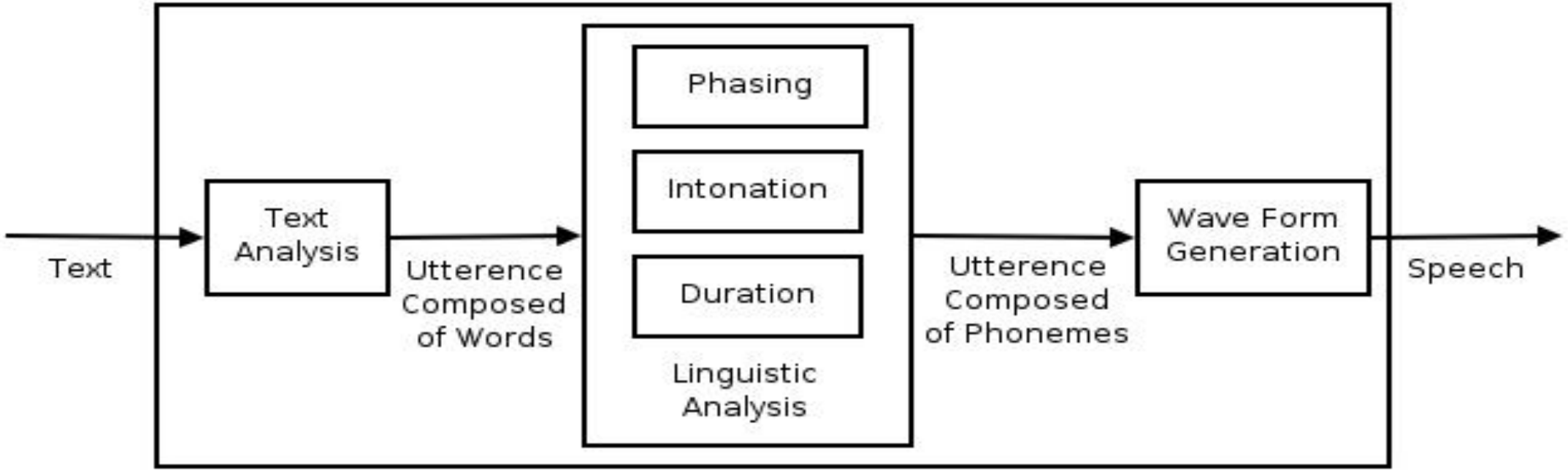
What is speech synthesis?

- The quality of a speech synthesizer is judged by its similarity to the human voice (*naturalness*), and by its ability to be understood (*intelligibility*).
- ***Naturalness*** describes how closely the output sounds like human speech, while ***intelligibility*** is the ease with which the output is understood
- An intelligible text-to-speech program allows people with visual impairments or reading disabilities to listen to written works on a home computer
- The ideal speech synthesizer is both natural and intelligible

How it works?

- The two primary technologies for generating synthetic speech waveforms are *concatenative synthesis* and *formant synthesis*
- A text-to-speech system (or "engine") is composed of two parts: a front-end and a back-end
- The front-end has two major tasks:
  - *text normalization:* it converts raw text containing symbols like numbers and abbreviations into the equivalent of written-out words
  - *grapheme-to-phoneme* conversion: it assigns phonetic transcriptions to each word, and divides and marks the text into prosodic units, like phrases, clauses and sentences
- The back-end (*synthesizer*): converts the symbolic linguistic representation into sound.

Overview of a typical TTS system

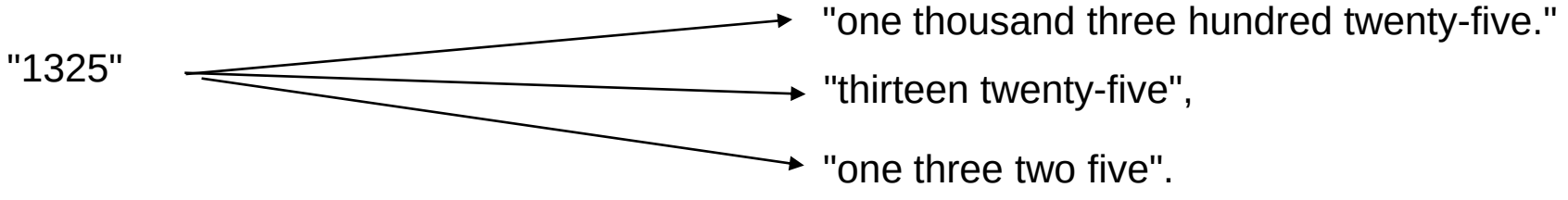


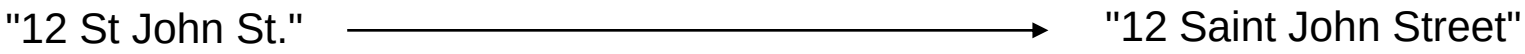
Synthesizer technologies

- Concatenative synthesis (based on the concatenation of segments of recorded speech):
  - Unit selection synthesis
  - Diphone synthesis
  - Domain-specific synthesis
- Formant synthesis (does not use human speech samples at runtime)
- Articulatory synthesis (refers to computational techniques for synthesizing speech based on models of the human vocal tract)
- HMM-based synthesis (based on hidden Markov models)

Text normalization challenges

- Texts are full of heteronyms, number and abbreviations that all require expansion into a phonetic representation
- Most text-to-speech (TTS) systems do not generate semantic representations of their input texts
- Various heuristic techniques are used to guess the proper way to disambiguate homographs, like examining neighboring words and using statistics about frequency of occurrence.
- Deciding how to convert numbers is another problem that TTS systems have to address
- Abbreviations can be ambiguous





Text-to-phoneme challenges

- dictionary-based approach:
  - a large dictionary containing all the words of a language and their correct pronunciations is stored by the program
  - determining the correct pronunciation of each word is a matter of looking up each word in the dictionary and replacing the spelling with the pronunciation specified in the dictionary
  - it's quick and accurate, but completely fails if it is given a word which is not in its dictionary

- rule-based approach:
  - pronunciation rules are applied to words to determine their pronunciations based on their spellings
  - works on any input, but the complexity of the rules grows substantially as the system takes into account irregular spellings or pronunciations

Voice synthesizers and API's

Microsoft Speech API

SoftVoice TTS

Apple PlainTalk

Nuance: Vocalizer

SVOX TTS

eSpeak

Applications

- Accessibility (speech synthesis has long been a vital assistive technology tool and its application in this area is significant and widespread)
- News service (some news sites used speech synthesis to convert written news to audio content, which can be used for mobile applications)
- Entertainment (Speech synthesis techniques are used as well in the entertainment productions such as games, animations)
- Navigation

Questions?

References

- Speech synthesis - http://en.wikipedia.org/wiki/Speech_synthesis
- Speech recognition - http://en.wikipedia.org/wiki/Speech_recognition
- SAPI - http://en.wikipedia.org/wiki/Speech_Application_Programming_Interface