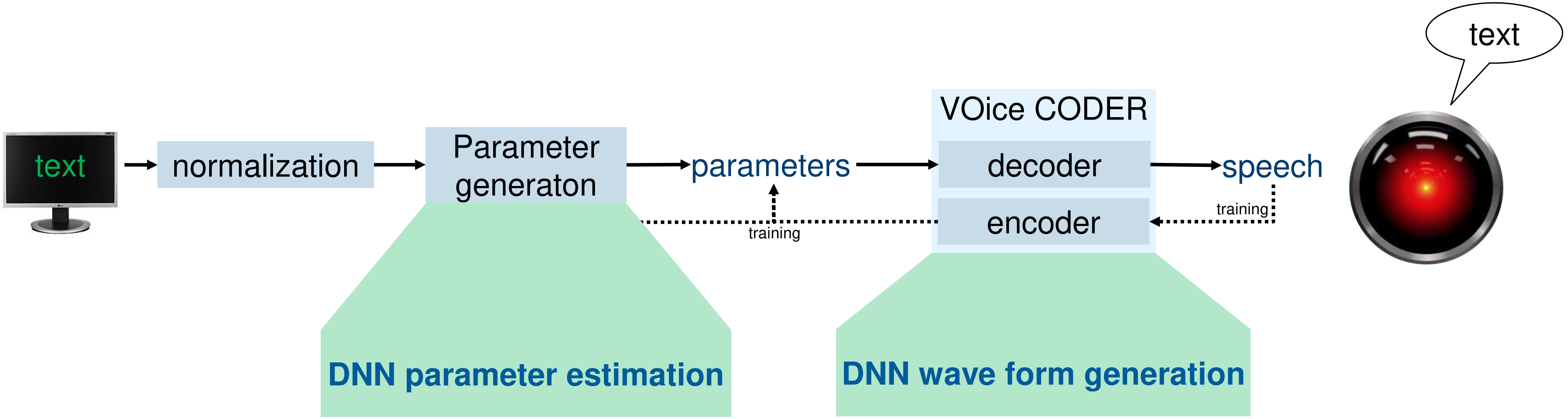
Text-to-Speech Synthesis

Parametric Synthesis



Text-to-Speech Synthesis

Parametric Synthesis

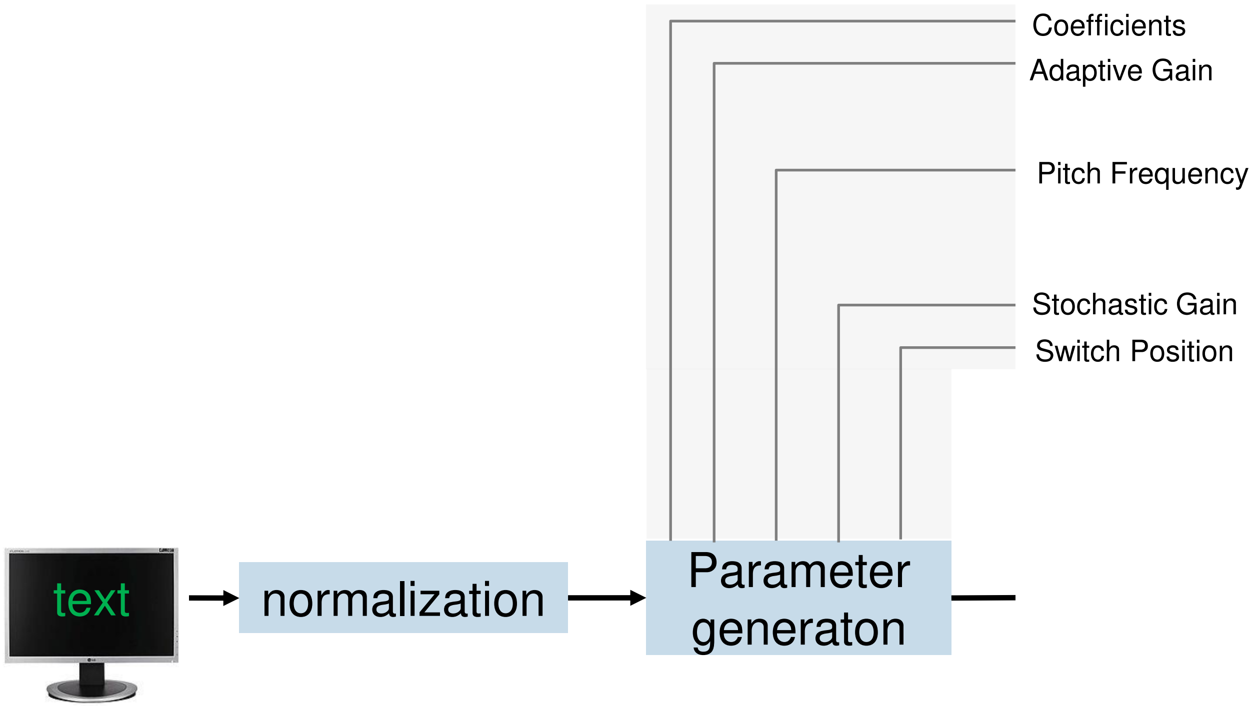
TTS-Standard
Format, e.g. w3c

→ normalization

Net patterns (email, web addresses)	Munir.George@THI.De
Date patterns	23/12/2021
Time patterns	10:24 h, 10:24
Duration patterns	11:12 h, 11 h 12 min
Currency patterns	8.95 €
Measure patterns	123.45 km
Telephone number patterns	+49 841 9348-2331
Number patterns (cardinal, ordinal, roman)	23rd III.
Abbreviations	Eng.
Special characters	&

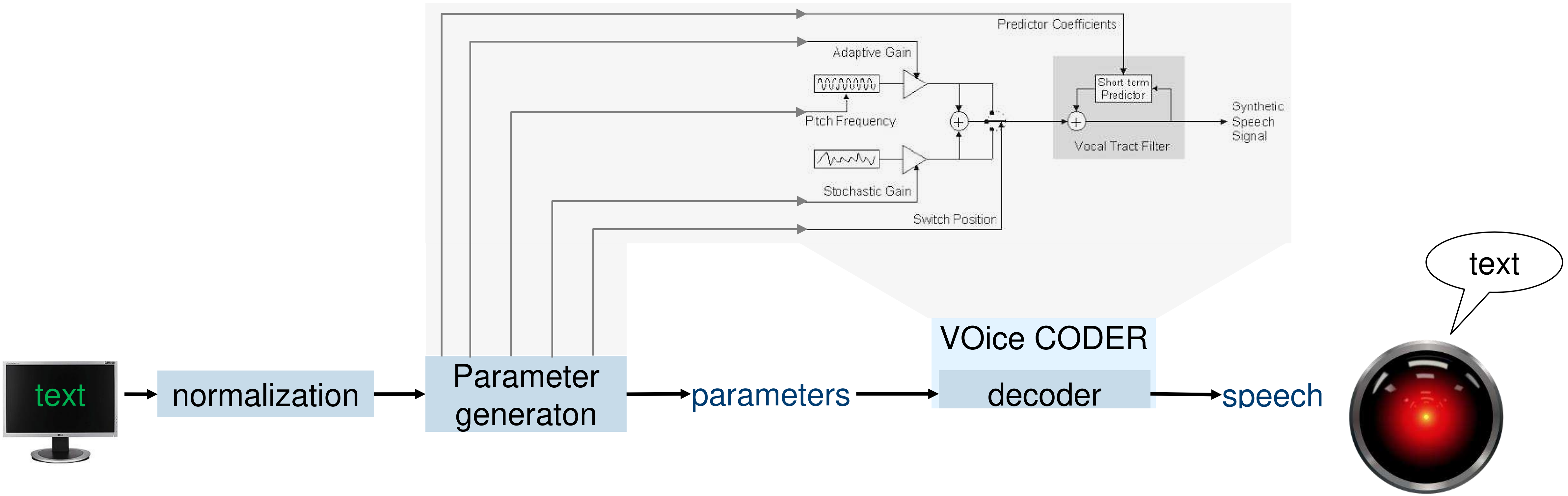
Text-to-Speech Synthesis

Parametric Synthesis



Text-to-Speech Synthesis

Parametric Synthesis with Most Common Vocoder (Voice coder): World



Widely used, complete framework: https://github.com/mmorise/World
https://www.jstage.jst.go.jp/article/transinf/E99.D/7/E99.D_2015EDP7457/_pdf/-char/en

Text-to-Speech Synthesis

Parametric Synthesis

First the text is processed to extract linguistic features, such as phonemes or duration. Second, it requires extraction of vocoder features, such as cepstra, spectrogram, fundamental frequency, etc., that represent some inherent characteristic of human speech, and are used in audio processing. These features are hand engineered and, along with the linguistic features are fed into a model called a Vocoder. While generating a waveform, the vocoder transforms the features and estimates parameters of speech like phase, speech rate, intonation.

Good things:

- Increased naturalness of the audio.
- Flexibility: it is easier to modify pitch for emotional change, or use MLLR adaptation to change voice characteristics;
- Lower development cost: it requires merely 2–3 hours of voice actor recording time which entangles less records, a smaller database and less data processing.

Text-to-Speech Synthesis

Parametric Synthesis

First the text is processed to extract linguistic features, such as phonemes or duration. Second, it requires extraction of vocoder features, such as cepstra, spectrogram, fundamental frequency, etc., that represent some inherent characteristic of human speech, and are used in audio processing. These features are hand engineered and, along with the linguistic features are fed into a model called a Vocoder. While generating a waveform, the vocoder transforms the features and estimates parameters of speech like phase, speech rate, intonation.

Good things:

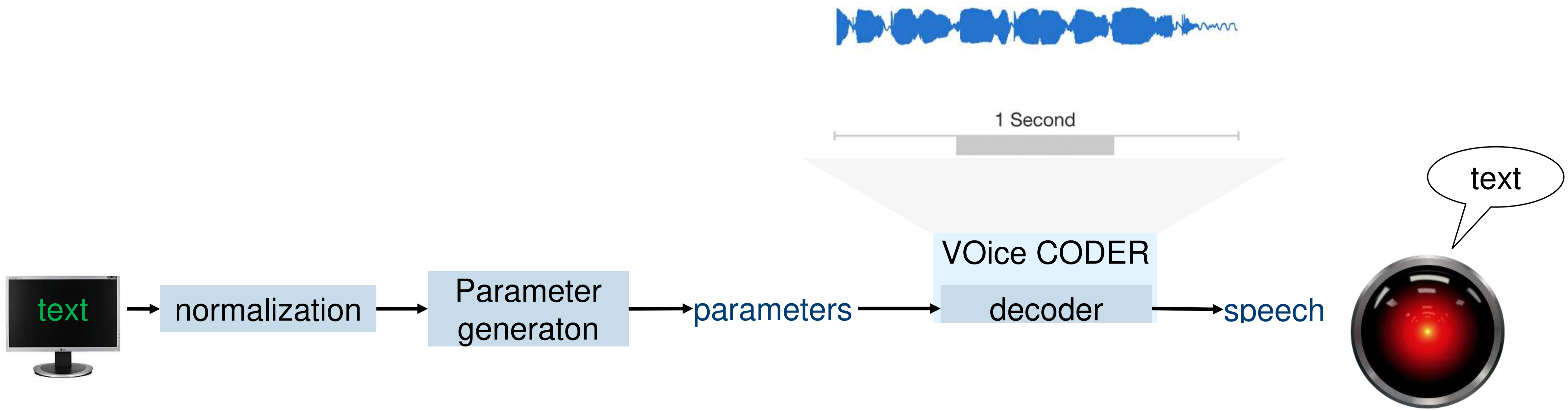
- Increased naturalness of the audio.
- Flexibility: it is easier to modify pitch for emotional change, or use MLLR adaptation to change voice characteristics;
- Lower development cost: it requires merely 2–3 hours of voice actor recording time which entangles less records, a smaller database and less data processing.

Bad things:

- Lower audio quality in terms of intelligibility: there are many artifacts resulting in muffled speech, with buzzing sound ever present, noisy audio;
- The voice can sound robotic: in the TTS based on a statistical model, the muffled sound makes the voice sound stable but unnatural and robotic.

Text-to-Speech Synthesis

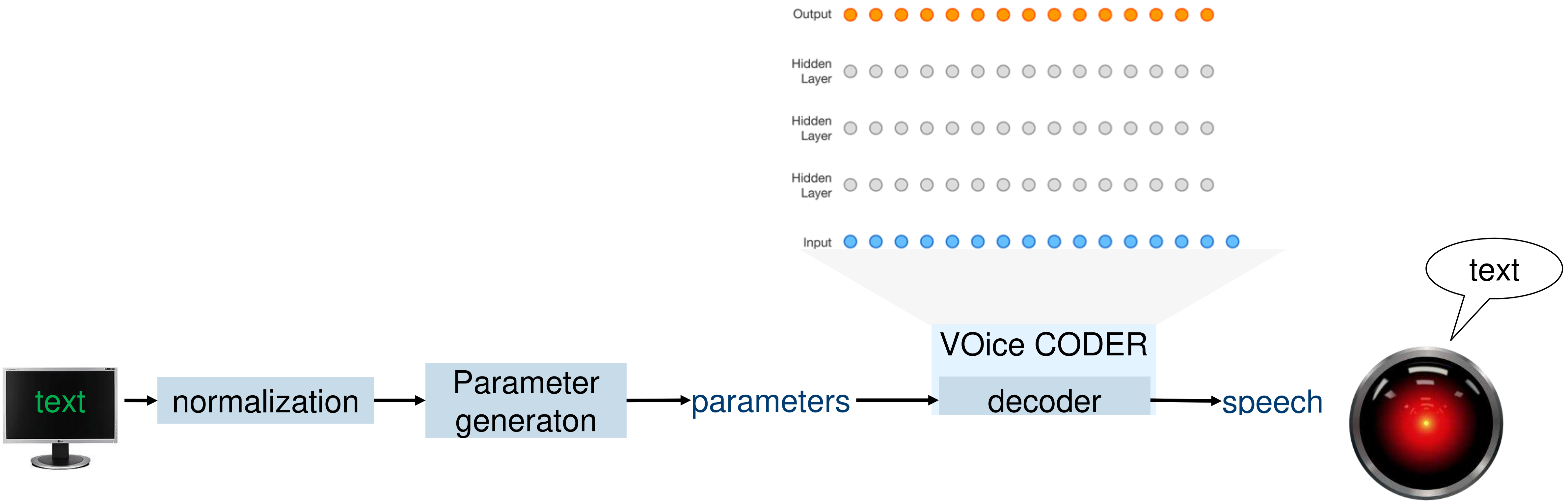
Alternative: Wavenet as vocoder



https://deepmind.com/blog/article/wavenet-generative-model-raw-audio
https://github.com/r9y9/wavenet_vocoder

Text-to-Speech Synthesis

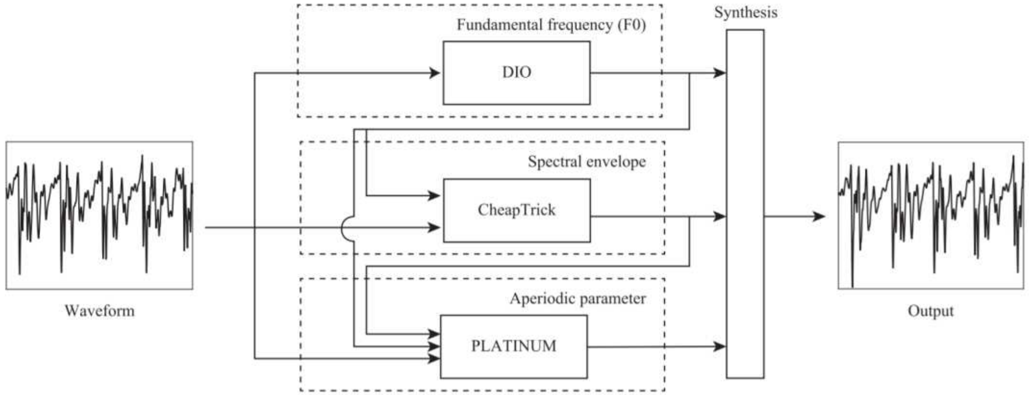
Alternative: Wavenet as vocoder



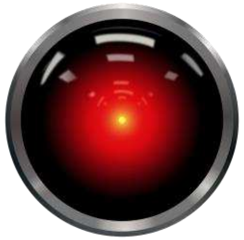
https://deepmind.com/blog/article/wavenet-generative-model-raw-audio
https://github.com/r9y9/wavenet_vocoder

Text-to-Speech Synthesis

Alternative: Griffin-Lim vocoder



→ normalization → Parameter generaton → parameters → VOice CODER decoder → speech

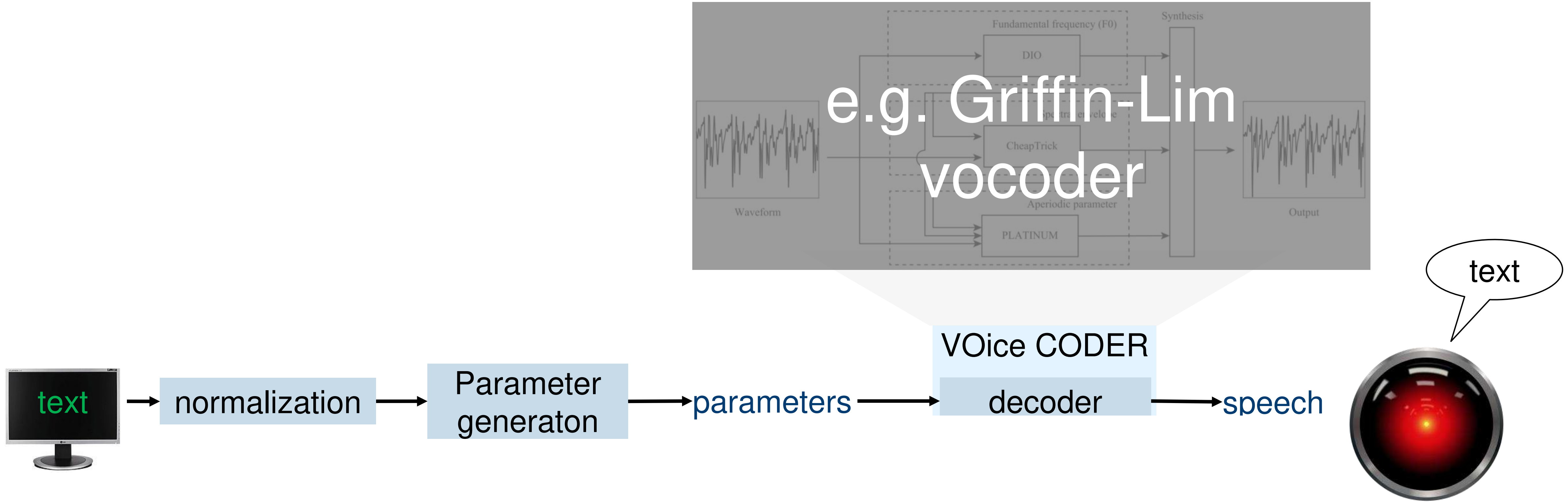


text

https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1164317

e.g., Applied by: http://web.stanford.edu/class/cs224s/reports/Alex_Barron.pdf

Alternative: Griffin-Lim vocoder



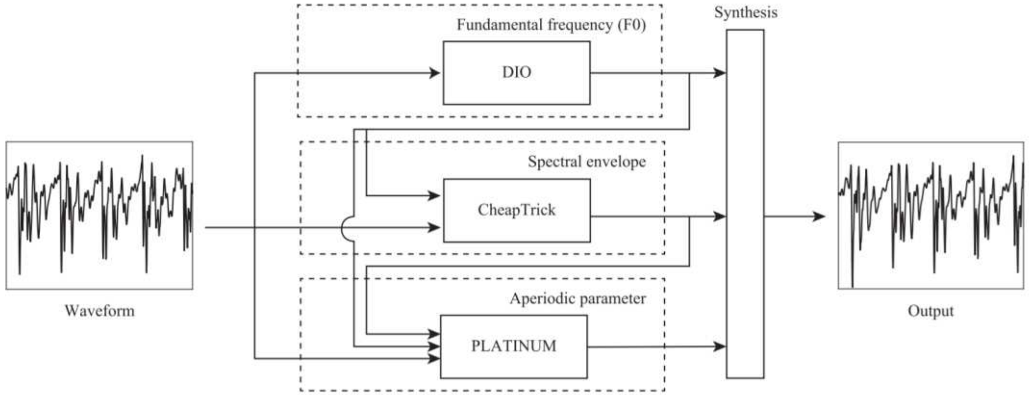
Griffin-Lim vocoder

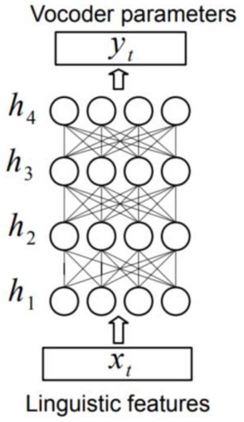
- **Invented in 1984 by Griffin and Lim**
- **It minimizes the mean squared error between a Short-Time Fourier Transform (STFT) of the estimated signal and the modified STFT**
- **Iterative algorithm to estimate a signal from its modified STFT magnitude**
- **It's differentiable: nice constrain as it enables to back propagate the gradients back to the DNN**

https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1164317
e.g., Applied by: http://web.stanford.edu/class/cs224s/reports/Alex_Barron.pdf

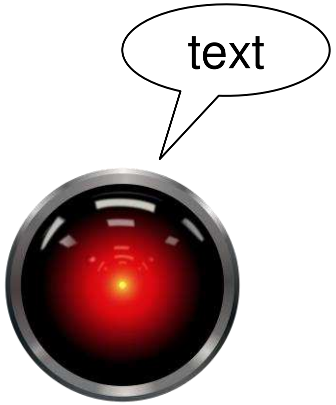
Text-to-Speech Synthesis

Merlin TTS (DNN-Vocoder Parameter estimation + world vocoder)





→ normalization → Parameter generaton → parameters → VOice CODER decoder → speech



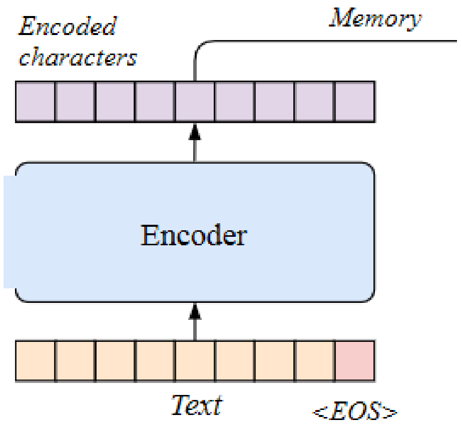
Vocoder parameters estimation:

- **6 feed forward NN layers with 1k hidden**
- **4 LSTM-RNN layer, 1k hidden + 1 LSTM-RNN with 0.5k hidden**
- **4 bi-LSTM-RNN layer, 0.384k hidden + 1 bi-LSTM-RNN with 0.384k hidden**

Widely used, complete framework: https://github.com/mmorise/World
https://www.jstage.jst.go.jp/article/transinf/E99.D/7/E99.D_2015EDP7457/_pdf/-char/en
http://ssw9.net/papers/ssw9_PS2-13_Wu.pdf

Text-to-Speech Synthesis

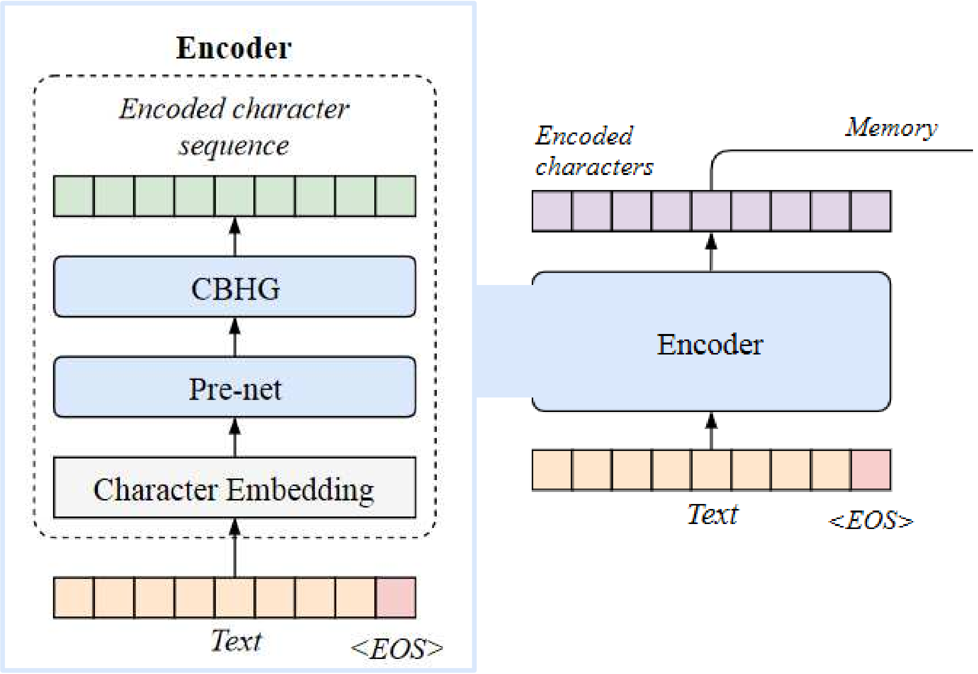
Encoder Decoder Architecture (Tacotron)



http://www.lab4inf.fh-muenster.de/lab4inf/docs/thesis/MA_weweler.pdf

Text-to-Speech Synthesis

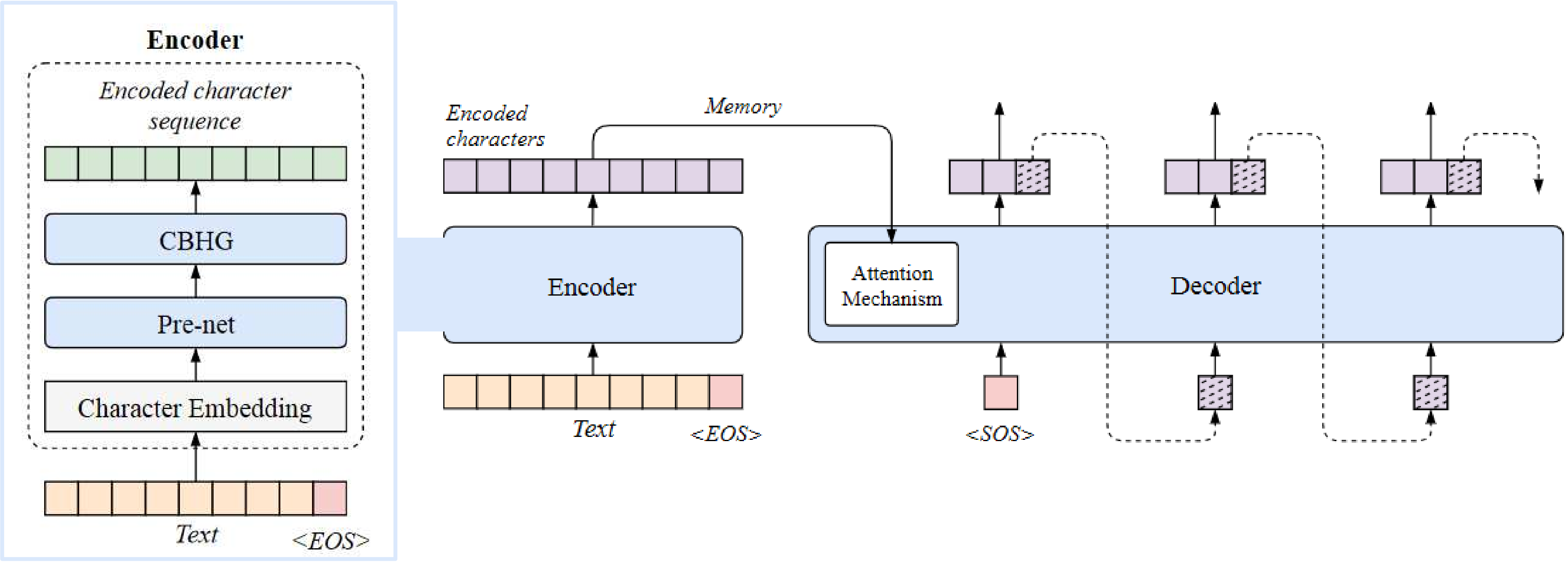
Encoder Decoder Architecture (Tacotron)



http://www.lab4inf.fh-muenster.de/lab4inf/docs/thesis/MA_weweler.pdf

Text-to-Speech Synthesis

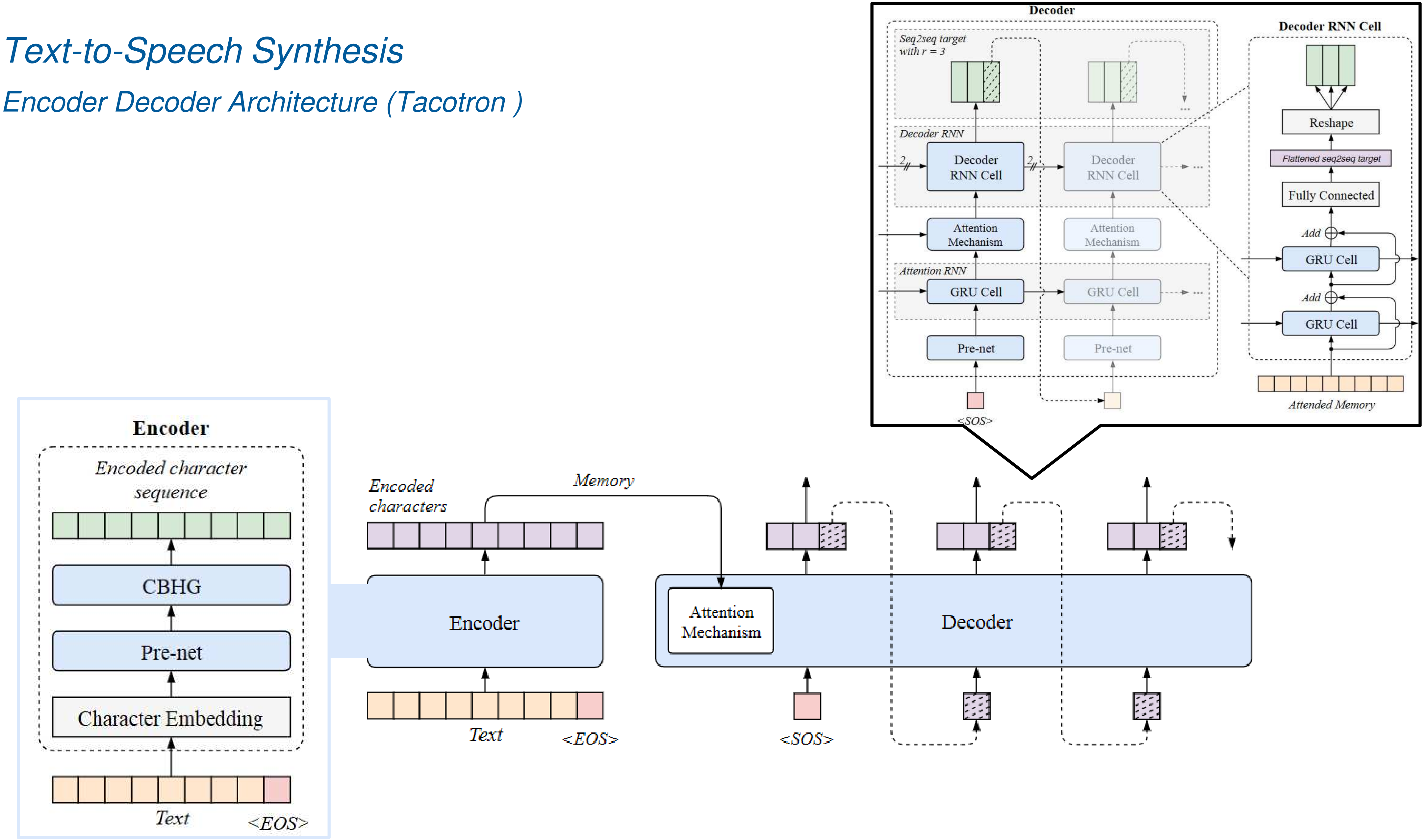
Encoder Decoder Architecture (Tacotron)



http://www.lab4inf.fh-muenster.de/lab4inf/docs/thesis/MA_weweler.pdf

Text-to-Speech Synthesis

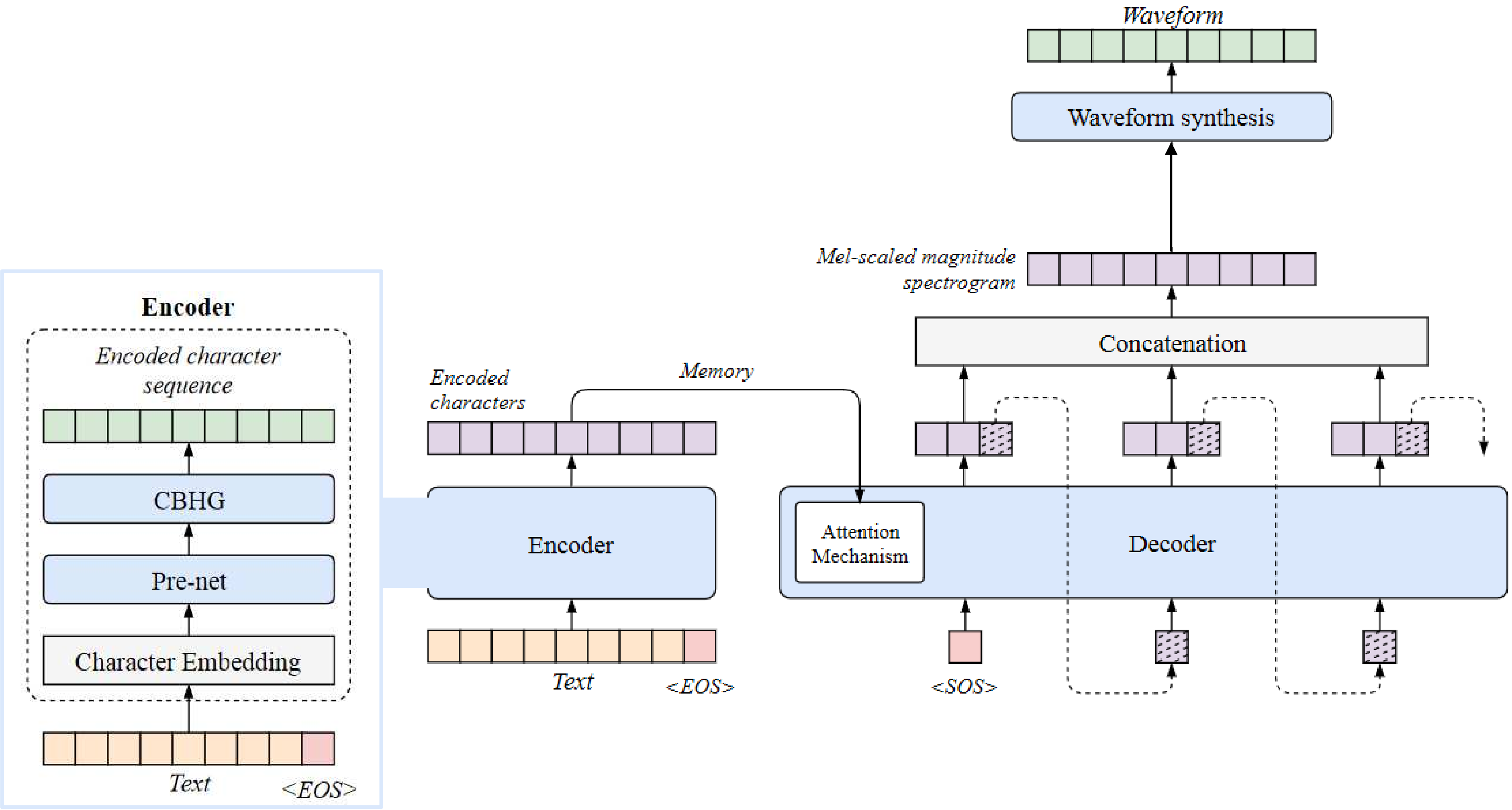
Encoder Decoder Architecture (Tacotron)



http://www.lab4inf.fh-muenster.de/lab4inf/docs/thesis/MA_weweler.pdf

Text-to-Speech Synthesis

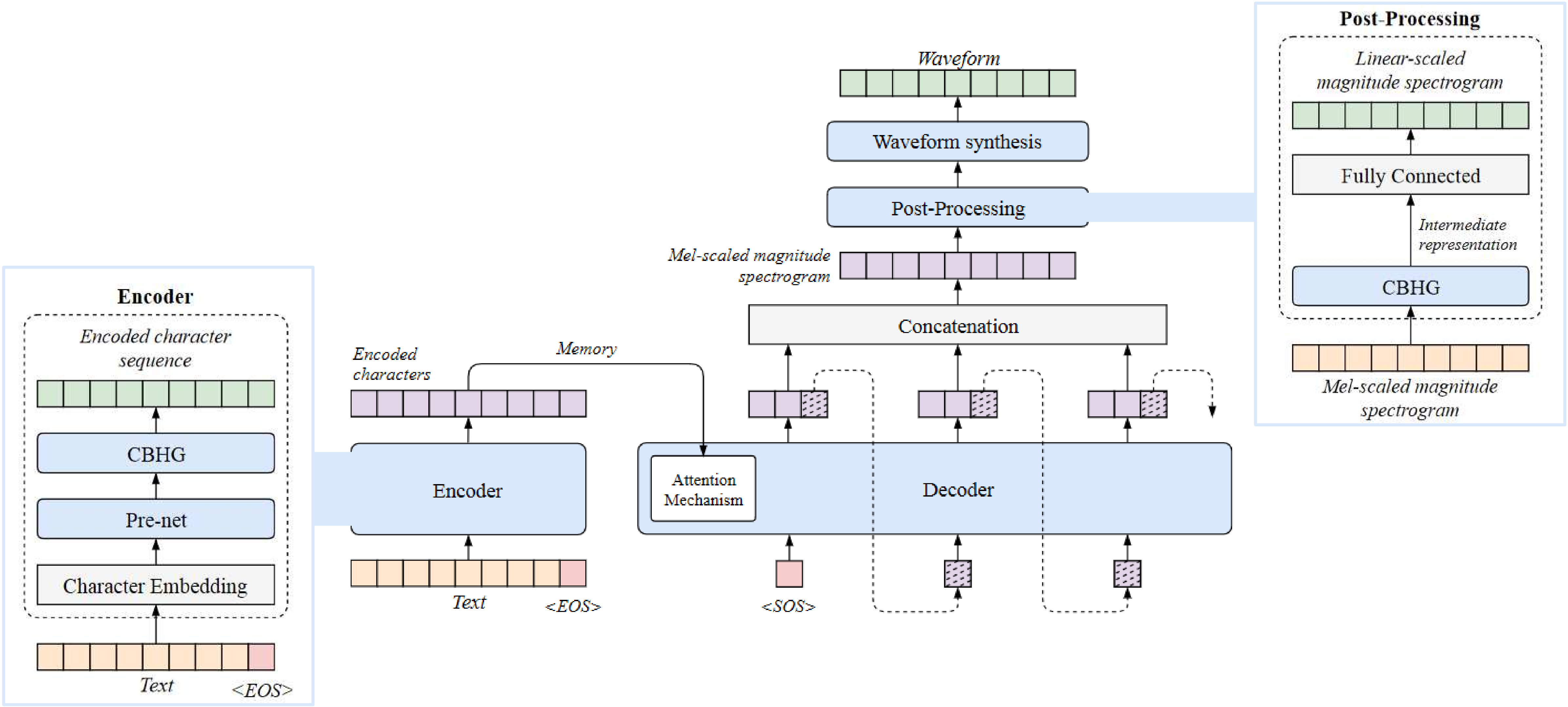
Encoder Decoder Architecture (Tacotron)



http://www.lab4inf.fh-muenster.de/lab4inf/docs/thesis/MA_weweler.pdf

Text-to-Speech Synthesis

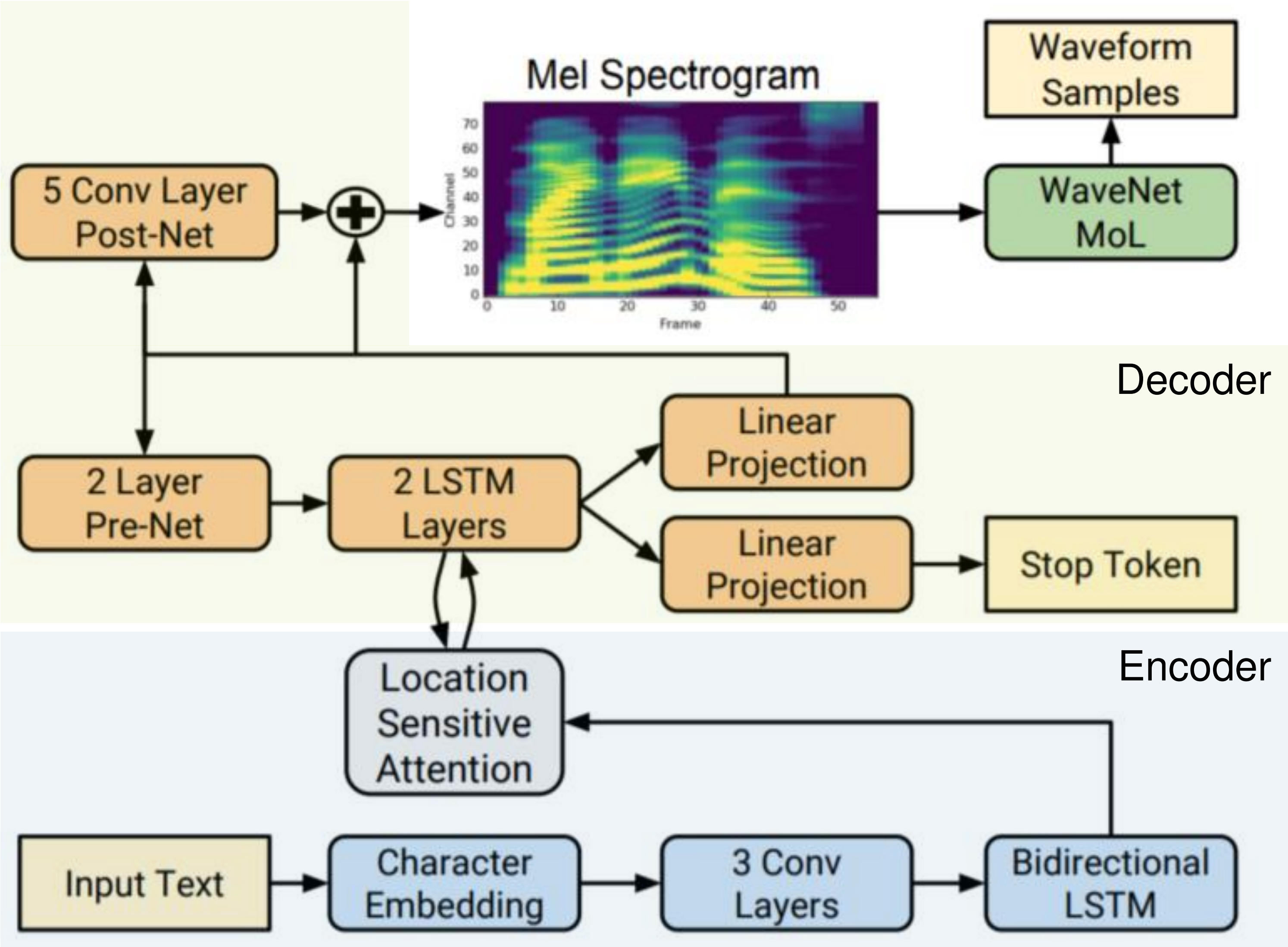
Encoder Decoder Architecture (Tacotron)



http://www.lab4inf.fh-muenster.de/lab4inf/docs/thesis/MA_weweler.pdf

Text-to-Speech Synthesis

Other example: Encoder Decoder Architecture (Tacotron 2)



https://arxiv.org/pdf/1712.05884.pdf
https://arxiv.org/pdf/1703.10135.pdf
https://arxiv.org/pdf/1803.09047.pdf
https://arxiv.org/pdf/1803.09017.pdf
https://github.com/NVIDIA/DeepLearningExamples/tree/master/PyTorch/SpeechSynthesis/Tacotron2

Other example: Deep Speech 3 TTS (DNN-Vocoder Parameter estimation + Griffin-Lim vocoder)

Encoder:

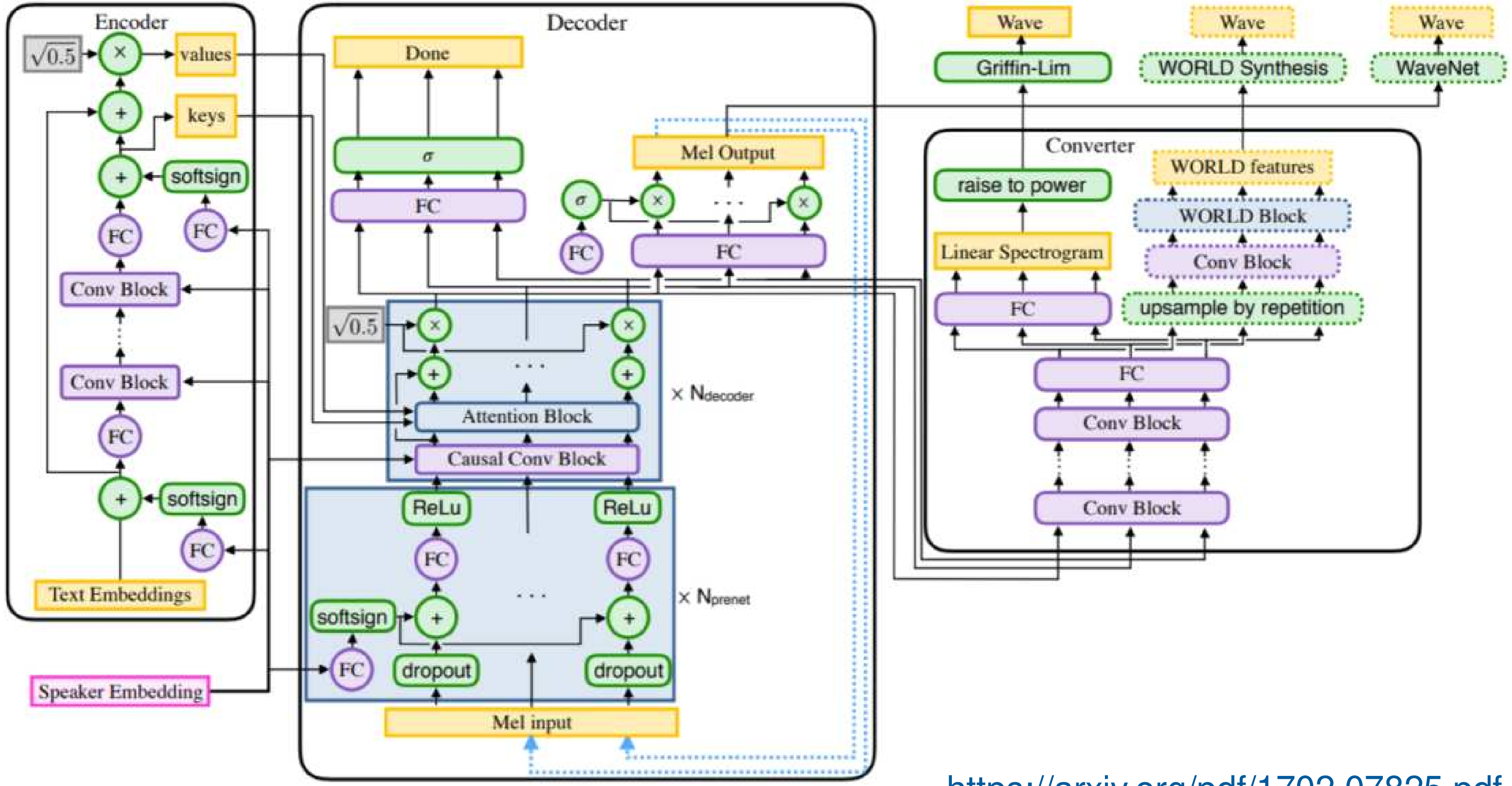
A fully-convolutional encoder, which converts textual features to an internal learned representation.

Decoder:

A fully-convolutional causal decoder, which decodes the learned representation with a multi-hop convolutional attention mechanism into a low-dimensional audio representation (mel-scale spectrograms) in an autoregressive manner.

Vocoder (Converter):

A fully-convolutional post-processing network, which predicts final vocoder parameters (depending on the vocoder choice) from the decoder hidden states. Unlike the decoder, the converter is non-causal and can thus depend on future context information.



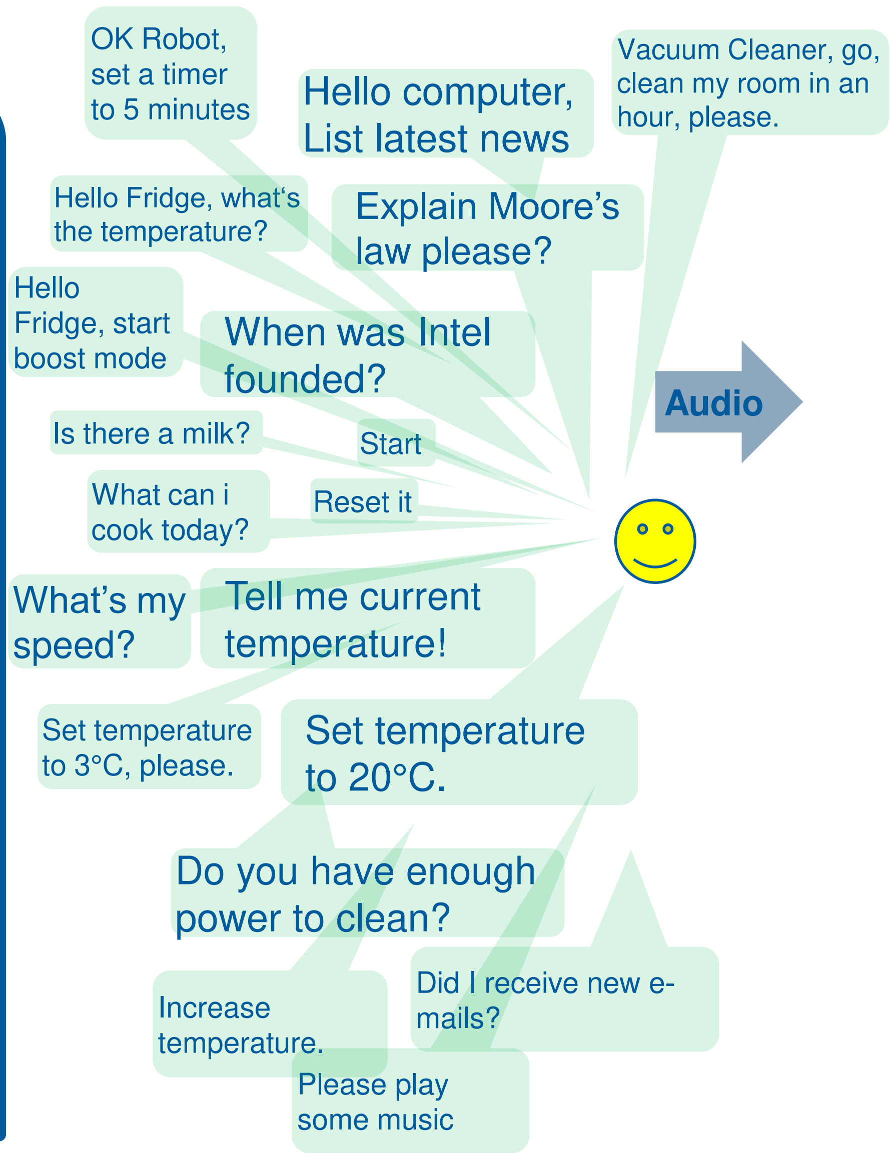
https://arxiv.org/pdf/1702.07825.pdf
https://arxiv.org/pdf/1710.07654.pdf
https://arxiv.org/pdf/1710.08969.pdf
https://arxiv.org/pdf/1705.08947.pdf

Dialog System Architecture

Overview

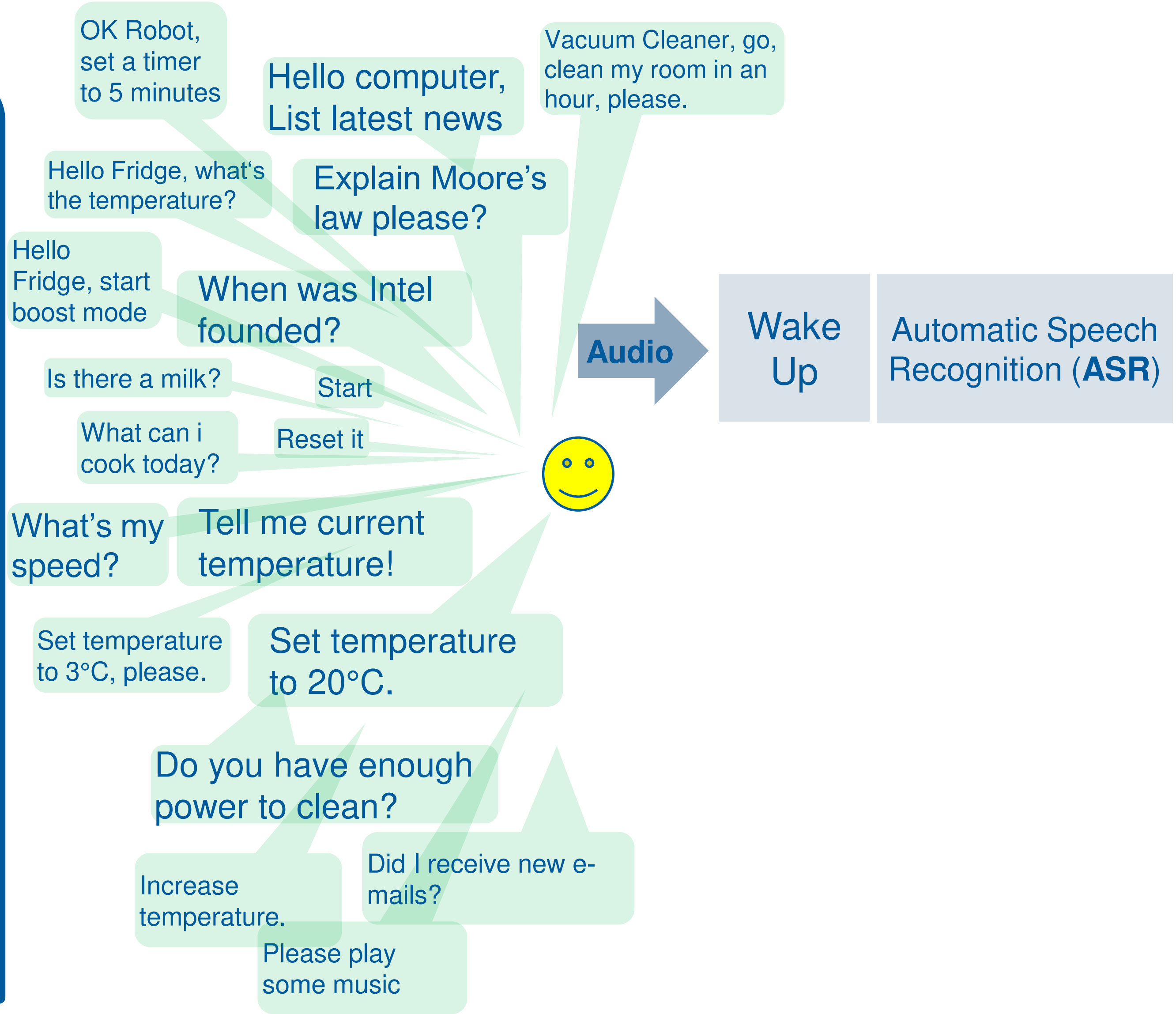
Dialog System Architecture

Overview



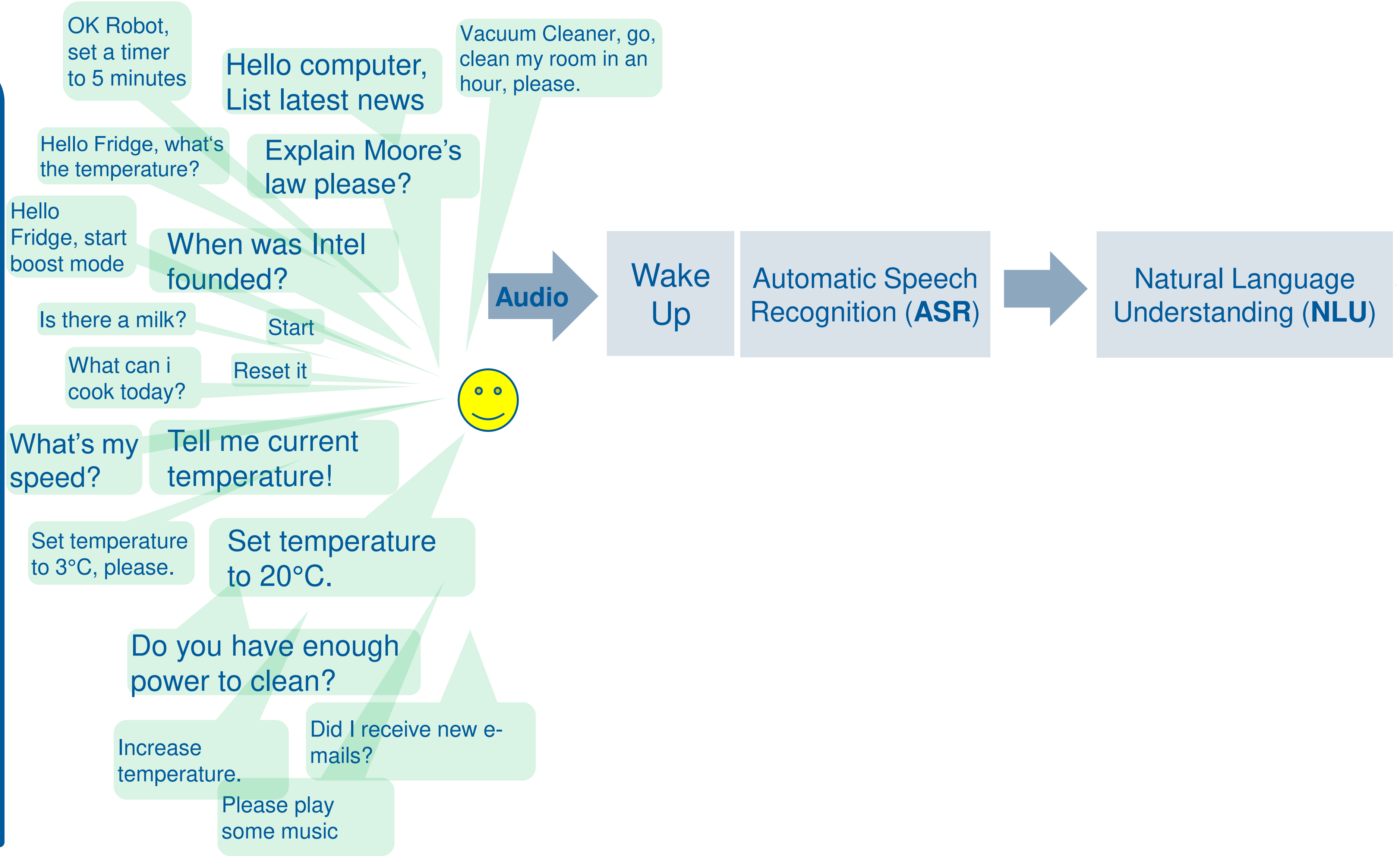
Dialog System Architecture

Overview



Dialog System Architecture

Overview



Dialog System Architecture

Overview

OK Robot, set a timer to 5 minutes

Hello computer, List latest news

Vacuum Cleaner, go, clean my room in an hour, please.

Hello Fridge, what's the temperature?

Explain Moore's law please?

Hello Fridge, start boost mode

When was Intel founded?

Is there a milk?

Start

What can i cook today?

Reset it

What's my speed?

Tell me current temperature!

Set temperature to 3°C, please.

Set temperature to 20°C.

Do you have enough power to clean?

Increase temperature.

Did I receive new e-mails?

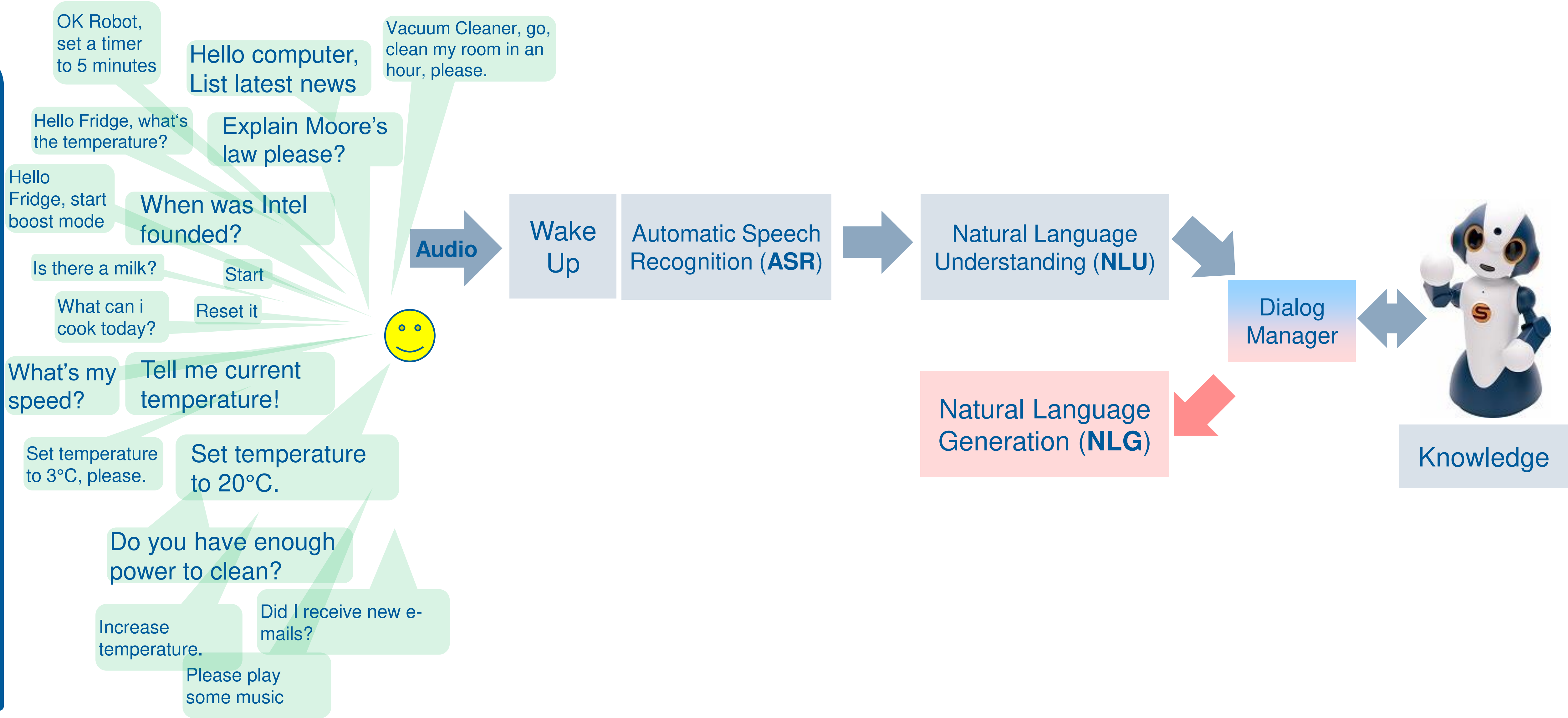
Please play some music

Audio → Wake Up → Automatic Speech Recognition (**ASR**) → Natural Language Understanding (**NLU**) → Dialog Manager ↔

Knowledge

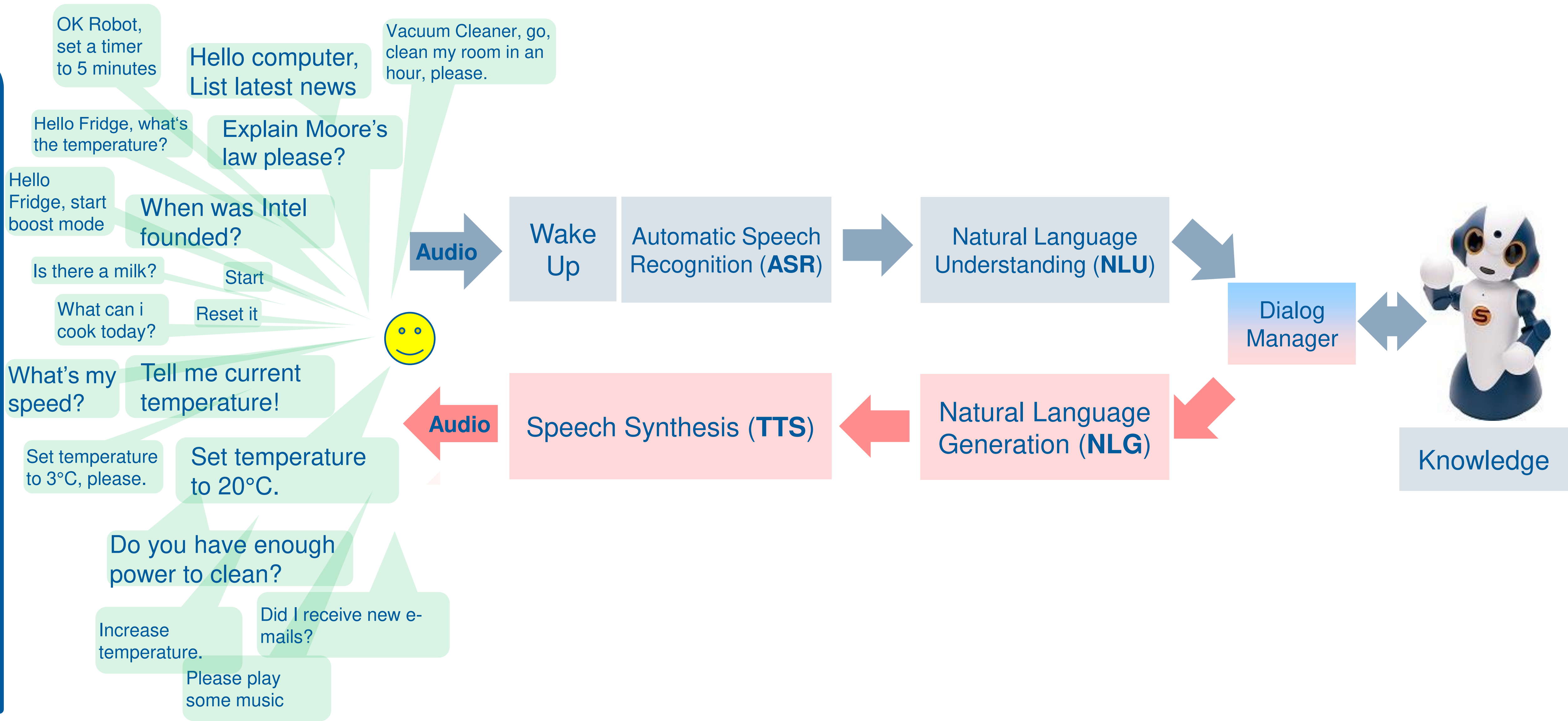
Dialog System Architecture

Overview



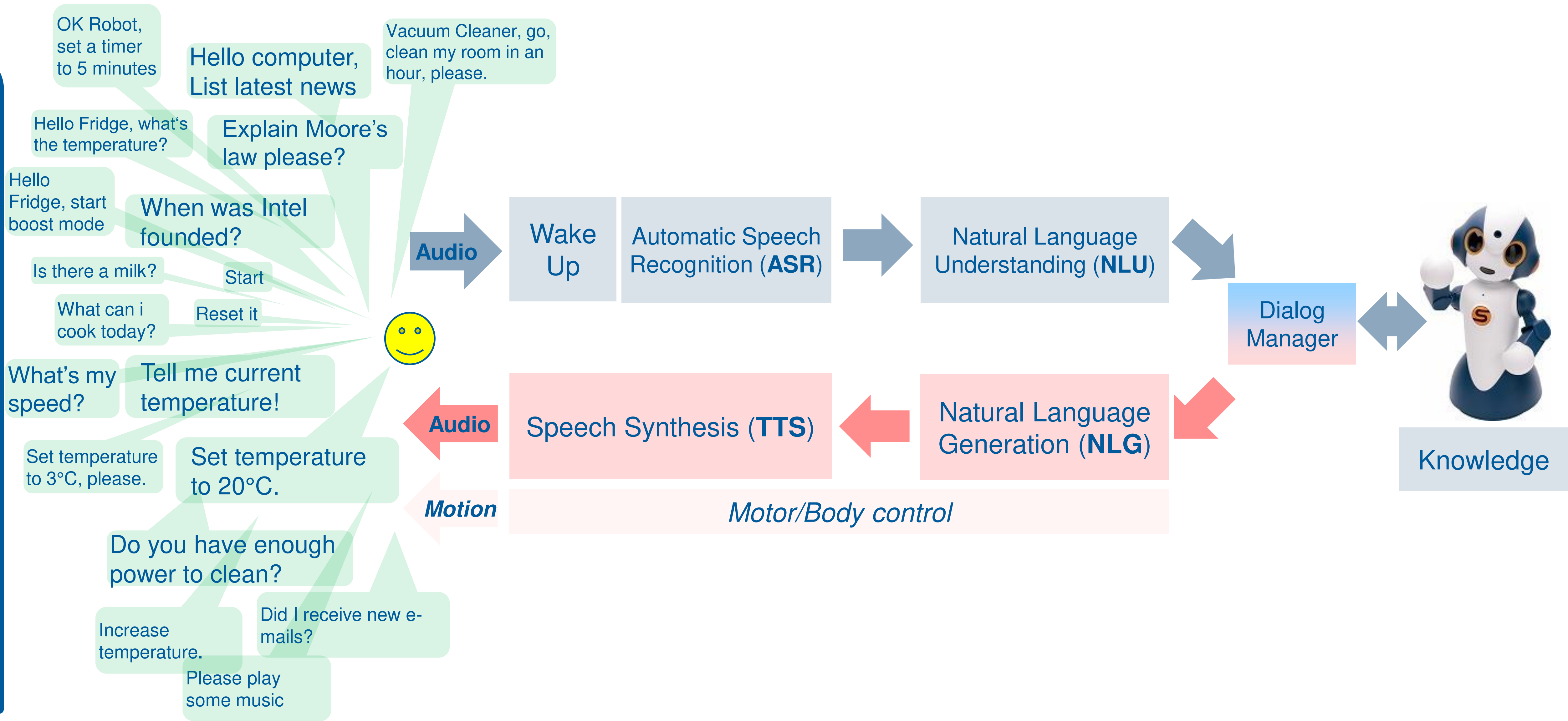
Dialog System Architecture

Overview



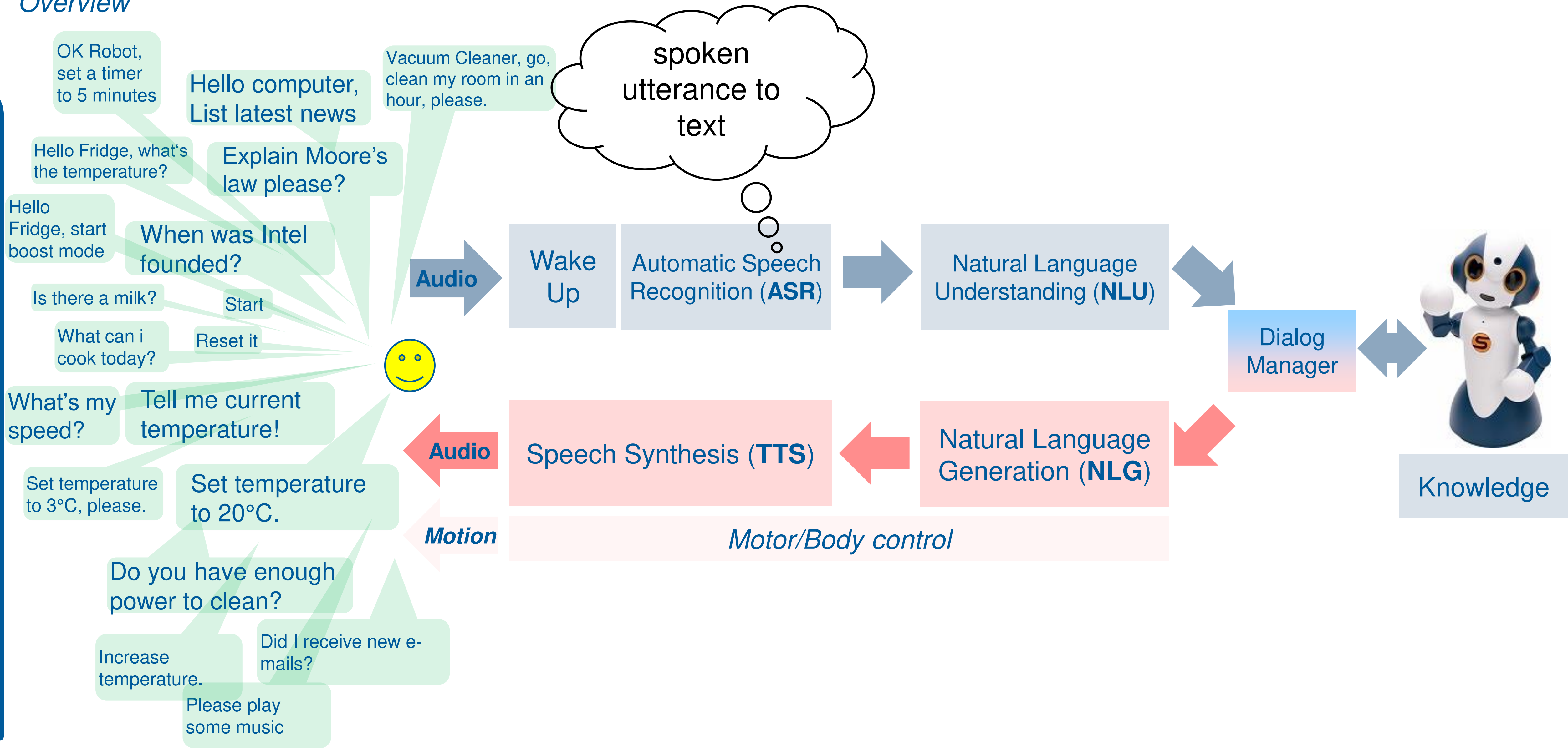
Dialog System Architecture

Overview



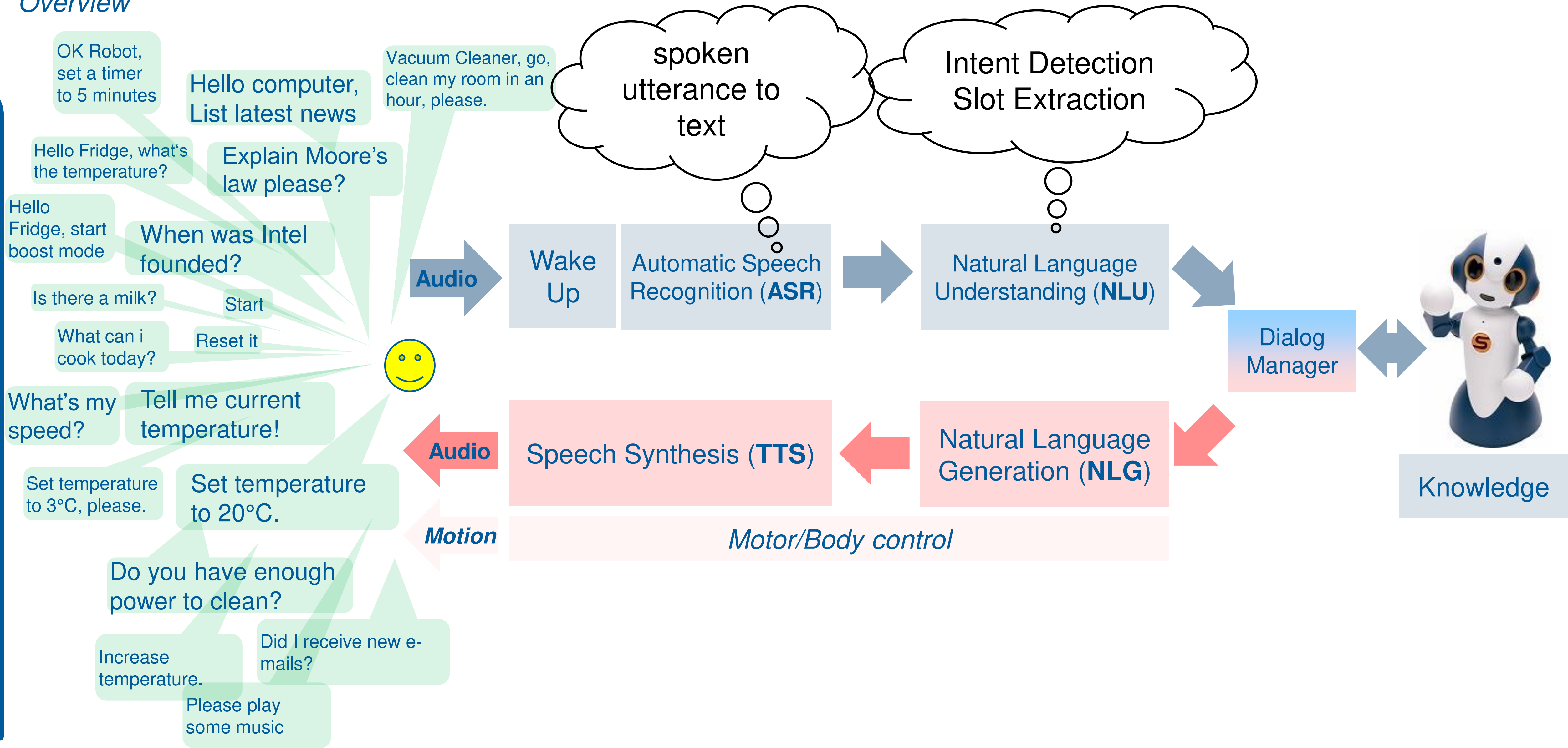
Dialog System Architecture

Overview



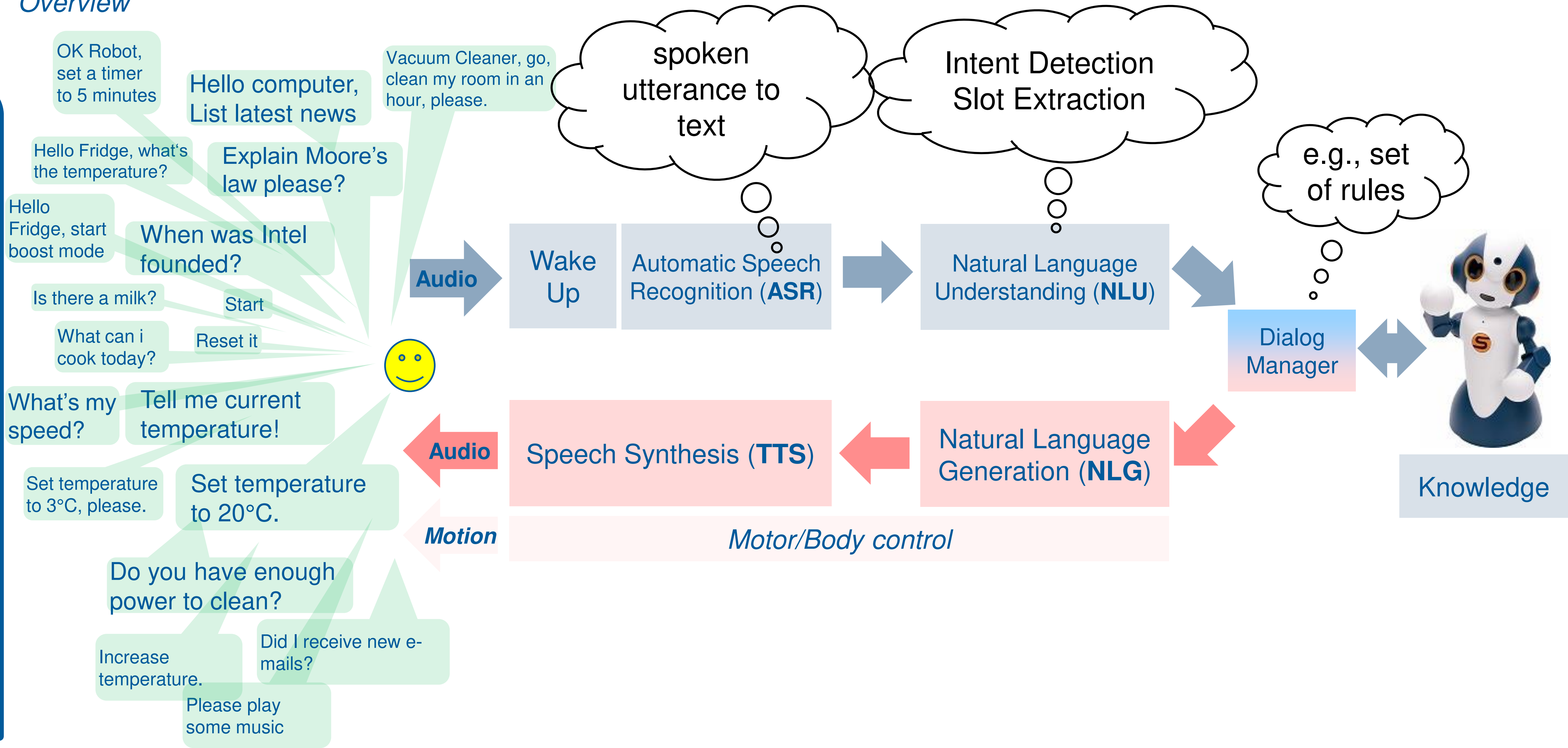
Dialog System Architecture

Overview



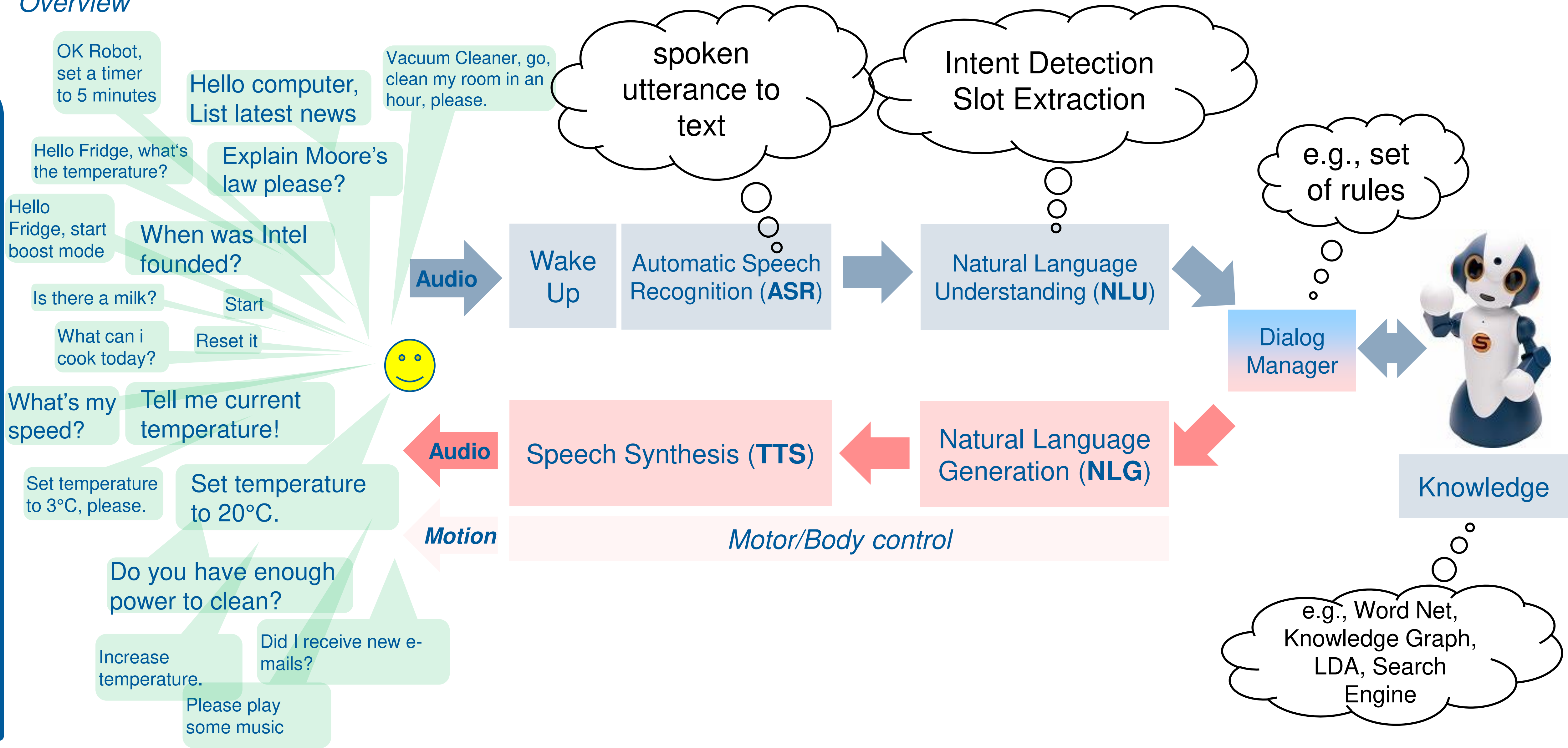
Dialog System Architecture

Overview



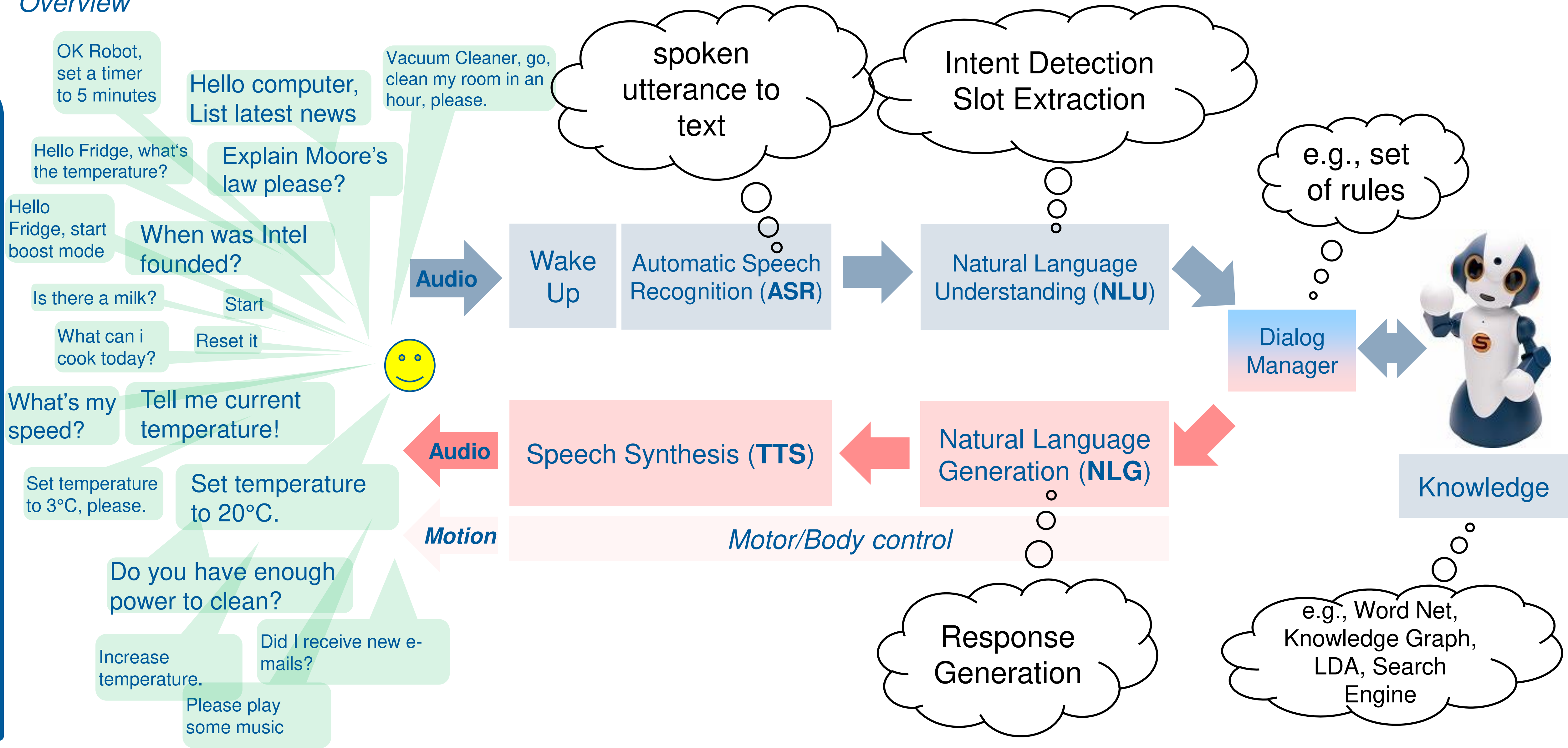
Dialog System Architecture

Overview



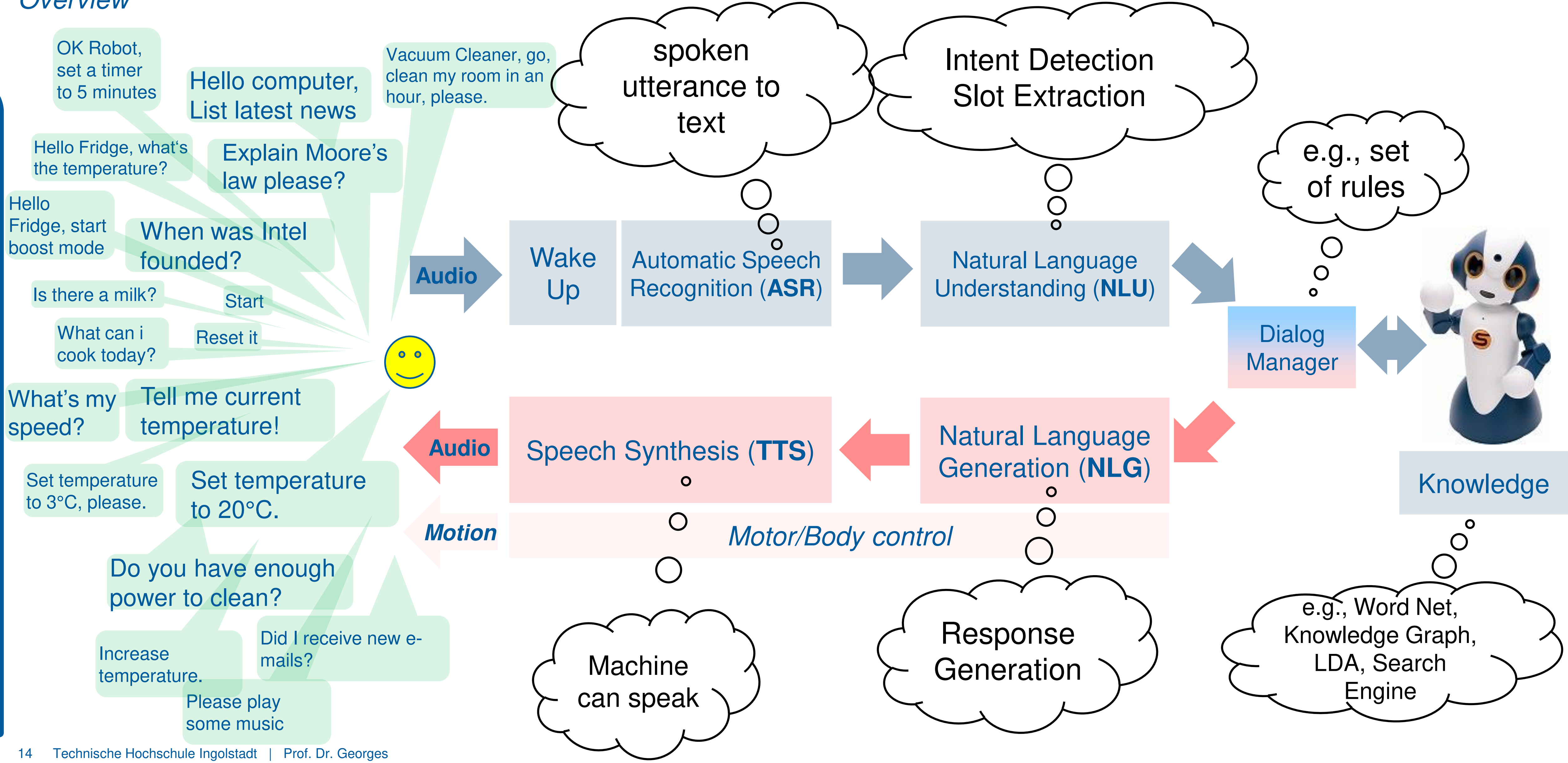
Dialog System Architecture

Overview



Dialog System Architecture

Overview



ELIZA (Weizenbaum, 1966) is the most important chatbot dialog system in the his-tory of the field, and so we take some time here to describe its algorithm in detail. ELIZA was designed to simulate a Rogerian psychologist, based on a branch of clinical psychology whose methods involve drawing the patient out by reflecting patient's statements back at them. Rogerian psychology is the rare type of conversation in which, as Weizenbaum points out, one can "assume the pose of knowing almost nothing of the real world".

```
USER:  Men are all alike
ELIZA: IN WHAT WAY
USER:  They're always bugging us about something or other.
ELIZA: CAN YOU THINK OF A SPECIFIC EXAMPLE
USER:  Well my boyfriend made me come here.
ELIZA: YOUR BOYFRIEND MADE YOU COME HERE
USER:  He says I'm depressed much of the time.
ELIZA: I AM SORRY TO HEAR YOU ARE DEPRESSED
…
ELIZA: WHO ELSE IN YOUR FAMILY TAKES CARE OF YOU?
USER:  My father
ELIZA: YOUR FATHER
USER:  You are like my father in some ways.
ELIZA: WHAT RESEMBLANCE DO YOU SEE
USER:  You are not very aggressive but I think you don't want me to notice that.
ELIZA: WHAT MAKES YOU THINK I AM NOT AGGRESSIVE
USER:  You don't argue with me.
ELIZA: WHY DO YOU THINK I DON'T ARGUE WITH YOU
USER:  You are afraid of me.
ELIZA: DOES IT PLEASE YOU TO BELIEVE I'M AFRAID OF YOU
USER:  My father is afraid of everybody.
ELIZA: WHAT ELSE COMES TO MIND WHEN YOU THINK OF YOUR FATHER
USER:  Bullies.
ELIZA: DOES THAT HAVE ANYTHING TO DO WITH THE FACT THAT YOUR BOYFRIEND MADE YOU COME HERE
```

Rule Based Dialog: ELIZA

ELIZA (Weizenbaum, 1966) is the most important chatbot dialog system in the his-tory of the field, and so we take some time here to describe its algorithm in detail. ELIZA was designed to simulate a Rogerian psychologist, based on a branch of clinical psychology whose methods involve drawing the patient out by reflecting patient's statements back at them. Rogerian psychology is the rare type of conversation in which, as Weizenbaum points out, one can "assume the pose of knowing almost nothing of the real world".

Example Rules:

```
(* YOU * ME)          ->(WHAT MAKES YOU THINK I 3 YOU)
(I *)                 ->(YOU SAY YOU 2 YOU)
```

pattern	transform

Example:

```
You love me                      -> WHAT MAKES YOU THINK I LOVE YOU
I know everybody laughed at me   -> YOU SAY YOU KNOW EVERYBODY LAUGHED AT YOU
```

Each utterance in a dialogue is a kind of action being performed by the speaker. These actions are commonly called *speech acts* or *dialog acts*.

Each utterance in a dialogue is a kind of action being performed by the speaker. These actions are commonly called *speech acts* or *dialog acts*.

Class of speech act	Description	Example
Constatives	committing the speaker to something's being the case	answering, claiming, confirming, denying, disagreeing, stating
Directives	attempts by the speaker to get the addressee to do something	advising, asking, forbidding, inviting, ordering, requesting
Commissives	committing the speaker to some future course of action	promising, planning, vowing, betting, opposing
Acknowledgements	express the speaker's attitude regarding the hearer with respect to some social action	apologizing, greeting, thanking, accepting an acknowledgment

Principle of closure. Agents performing an action require evidence, sufficient for current purposes, that they have succeeded in performing it.

Principle of closure. Agents performing an action require evidence, sufficient for current purposes, that they have succeeded in performing it.

- **Need to know whether an action succeeded or failed**

Principle of closure. Agents performing an action require evidence, sufficient for current purposes, that they have succeeded in performing it.

- **Need to know whether an action succeeded or failed**
- **Dialogue is also an action**

Principle of closure. Agents performing an action require evidence, sufficient for current purposes, that they have succeeded in performing it.

- **Need to know whether an action succeeded or failed**
- **Dialogue is also an action**
  - A collective action performed by speaker and hearer
  - Common ground: set of things mutually believed by both speaker and hearer

Principle of closure. Agents performing an action require evidence, sufficient for current purposes, that they have succeeded in performing it.

- **Need to know whether an action succeeded or failed**
- **Dialogue is also an action**
  - A collective action performed by speaker and hearer
  - Common ground: set of things mutually believed by both speaker and hearer
- **Need to achieve common ground, so hearer must ground or acknowledge speakers utterance**

How do speakers ground?

How do speakers ground?

- **Continued attention:** B continues attending to A

How do speakers ground?

- **Continued attention:** B continues attending to A
- **Relevant next contribution:** B starts in on next relevant contribution

How do speakers ground?

- **Continued attention:** B continues attending to A
- **Relevant next contribution:** B starts in on next relevant contribution
- **Acknowledgement:** B nods or says continuer („uh-huh“) or assessment („great!“)

How do speakers ground?

- **Continued attention:** B continues attending to A
- **Relevant next contribution:** B starts in on next relevant contribution
- **Acknowledgement:** B nods or says continuer („uh-huh“) or assessment („great!“)
- **Demonstration:**

  B demonstrates understanding A by reformulating A‘s contribution, or by collaboratively completing A‘s utterance

How do speakers ground?

- **Continued attention:** B continues attending to A
- **Relevant next contribution:** B starts in on next relevant contribution
- **Acknowledgement:** B nods or says continuer („uh-huh“) or assessment („great!“)
- **Demonstration:**

  B demonstrates understanding A by reformulating A‘s contribution, or by collaboratively completing A‘s utterance
- **Display:**

  B repeats verbatim all or part of A‘s presentation

- **Display:**

  C: I need to travel in May.

  A: And, what day in May did you want to travel?

- **Acknowledgement:**

  C: I want to fly from Boston.

  A: mm-hmm.

  C: to Baltimore Washington International.

- **Display:**

  C: I need to travel in May.

  A: And, what day in May did you want to travel?

  Indicates to client that agent has successfully understood answer to the last question

- **Acknowledgement:**

  C: I want to fly from Boston.

  A: mm-hmm.

  C: to Baltimore Washington International.

- **Display:**

  C: I need to travel in May.

  A: And, what day in May did you want to travel? → Next relevant contribution

- **Acknowledgement:**

  C: I want to fly from Boston.

  A: mm-hmm.

  C: to Baltimore Washington International.

- **Display:**

  C: I need to travel in May.

  A: And, what day in May did you want to travel? → Next relevant contribution

  … you‘re flying into what city?
  … what time would you like to leave?

- **Acknowledgement:**

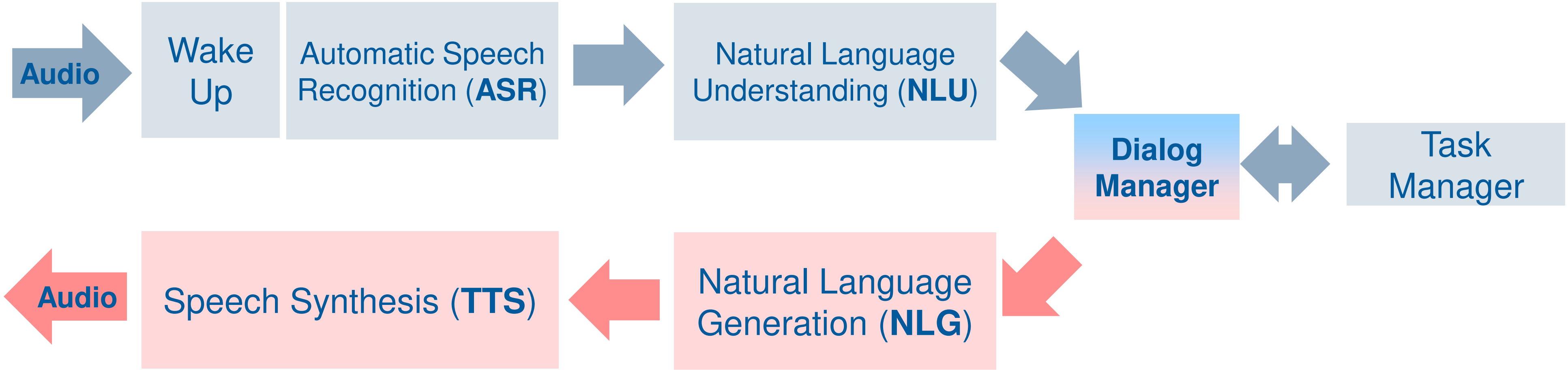
  C: I want to fly from Boston.

  A: mm-hmm.

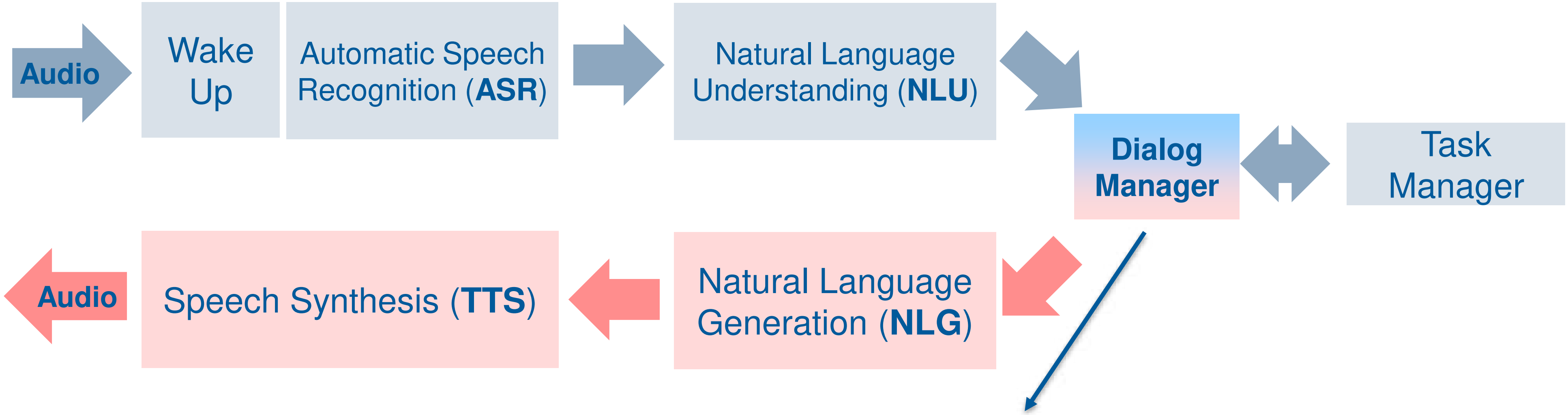
  C: to Baltimore Washington International.

Chatbot/Dialog Systems

Dialog Manager



Dialog Manager

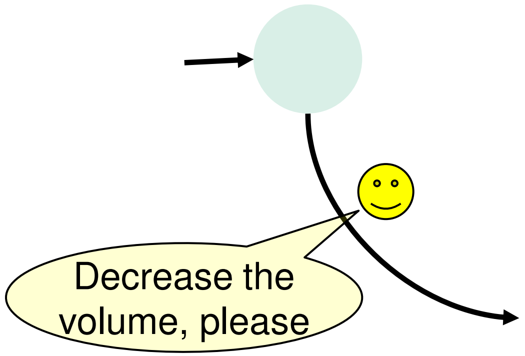


- **Controls architecture and structure of dialog**
  - Takes input from ASR/NLU components
  - Maintains some sort of *state*
  - Interfaces with *task manager*
  - Passes output to NLG/TTS modules

- Often we whink of simpler dialog tasks as interactively completing a data structure or **frame**
- Task execution (e.g. making a reservation) can happen via APIs etc.
- Defining the data structure required to complete a task can be difficult and time consuming
- Some modern approaches attempt to learn dialog/task actions directly (e.g. simulate clicks or API calls made by a human agent

- **Finite State**
- **Frame-based** (Alexa skills kit uses a version of this)
- **Information State (Markov Decision Process)**
- **Distributional / Neural network**

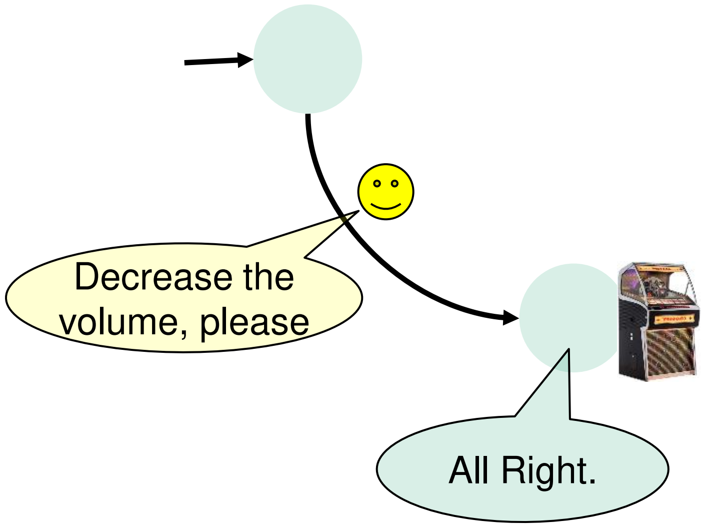
- **Intents: Yes, No, Music.On, Music.Decrease, …**



Chatbot/Dialog Systems

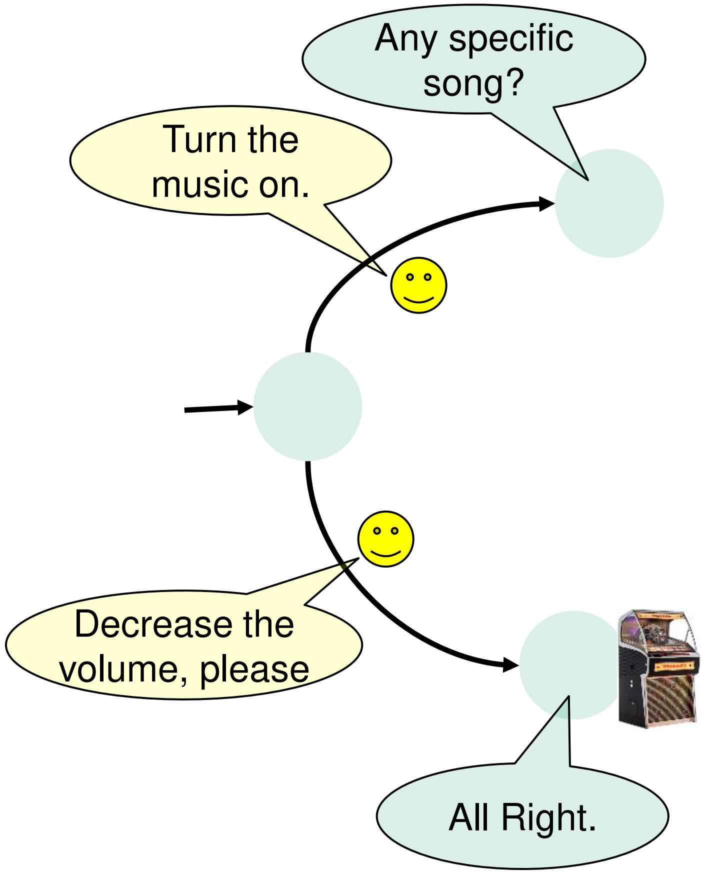
Rule Based Dialog

- **Intents: Yes, No, Music.On, Music.Decrease, …**
- **Response Generation: “Enjoy”, “All Right”, …**



Rule Based Dialog

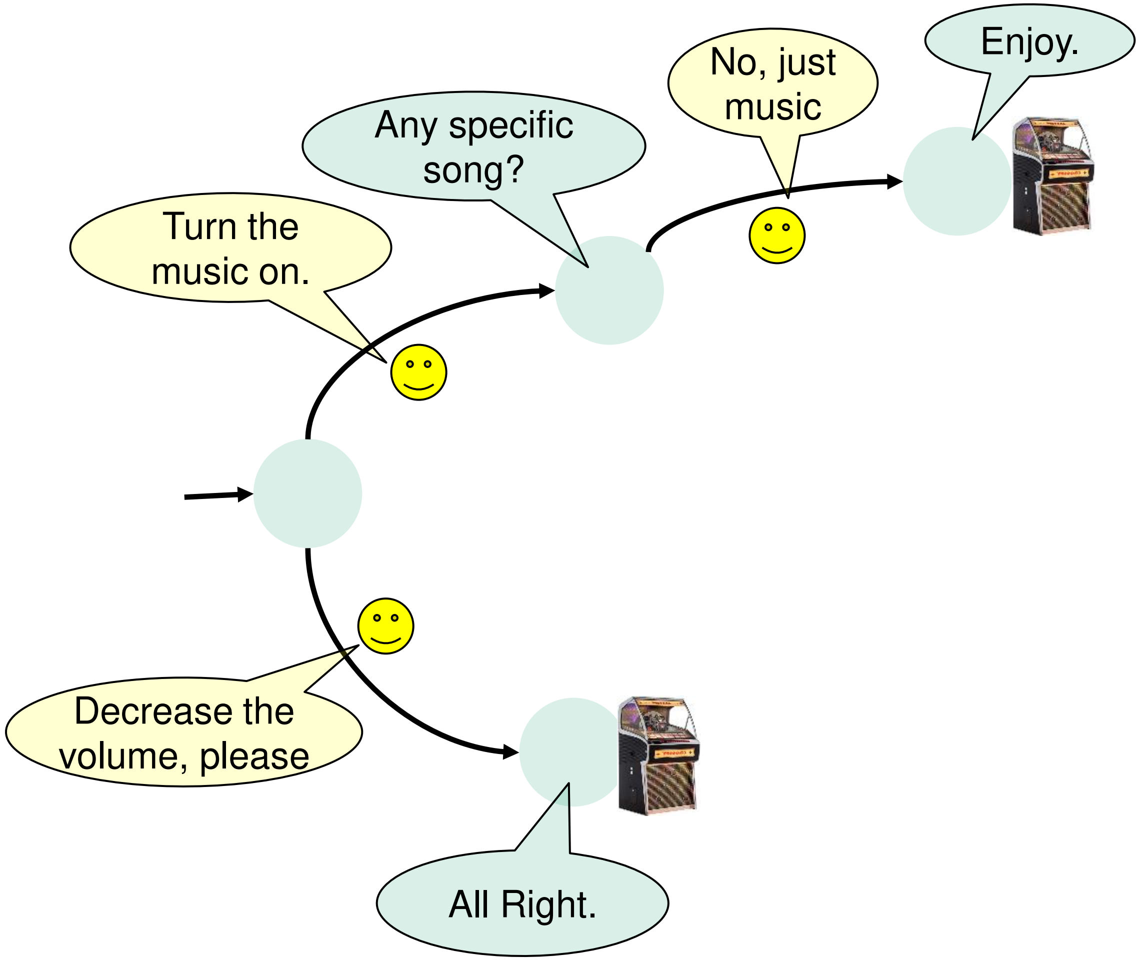
- **Intents: Yes, No, Music.On, Music.Decrease, …**
- **Response Generation: “Enjoy”, “All Right”, …**



Chatbot/Dialog Systems

Rule Based Dialog

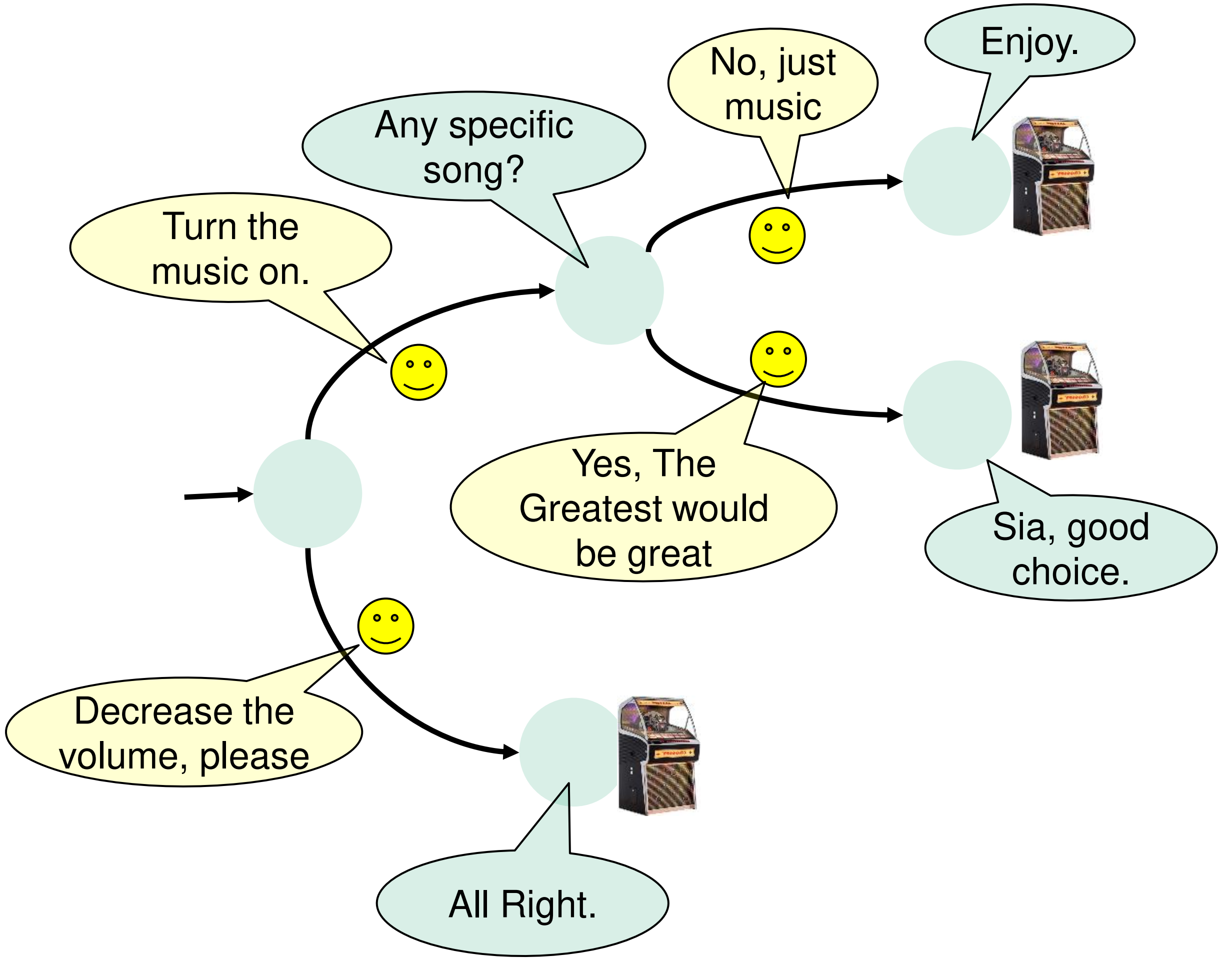
- **Intents: Yes, No, Music.On, Music.Decrease, …**
- **Response Generation: “Enjoy”, “All Right”, …**



Chatbot/Dialog Systems

Rule Based Dialog

- **Intents: Yes, No, Music.On, Music.Decrease, …**
- **Slots: “The Greatest”, …**
- **Response Generation: “Enjoy”, “All Right”, …**



Corpus-based chatbots**, instead of using hand-built rules, mine conversations of human-human conversations. These systems are enormously data-intensive, requiring hundreds of millions or even billions of words for training.**

***Corpus-based chatbots*, instead of using hand-built rules, mine conversations of human-human conversations. These systems are enormously data-intensive, requiring hundreds of millions or even billions of words for training.**

- **Retrieval methods**
- **Generation methods**

Empathetic Dialogues: https://aclanthology.org/P19-1534/

Using information retrieval to grab a response from

some corpus that is appropriate given the dialogue context

Using information retrieval to grab a response from some corpus that is appropriate given the dialogue context

- think of the **user's turn** as a **query q,**
- **Goal:** retrieve and repeat some appropriate **turn r** as the response from a corpus of **conversations C**

Using information retrieval to grab a response from some corpus that is appropriate given the dialogue context

- think of the **user's turn** as a **query q,**
- **Goal:** retrieve and repeat some appropriate **turn r** as the response from a corpus of **conversations C**

$$\text{response}(q, C) = \underset{r \in C}{\operatorname{argmax}} \frac{q \cdot r}{|q||r|}$$

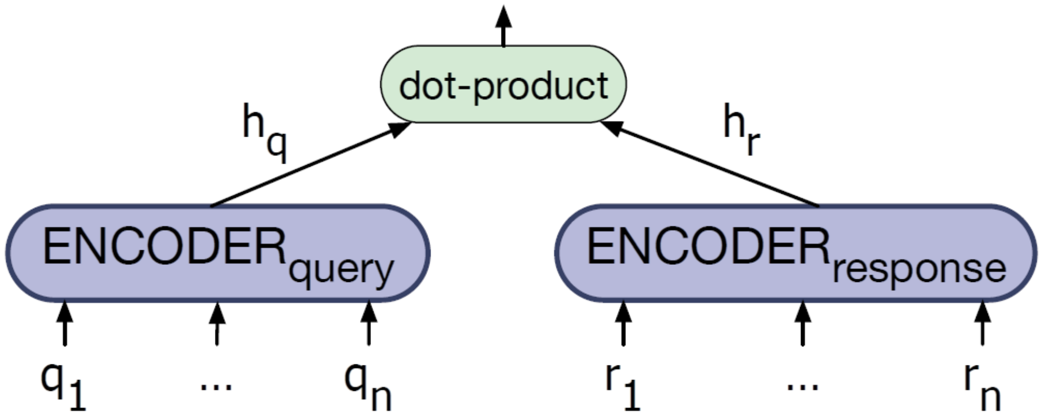
Using information retrieval to grab a response from some corpus that is appropriate given the dialogue context

- think of the **user’s turn** as a **query q,**
- **Goal:** retrieve and repeat some appropriate **turn r** as the response from a corpus of **conversations C**

$$
\begin{aligned}
h_q &= \text{BERT}_Q(\text{q})\,[\text{CLS}] \\
h_r &= \text{BERT}_R(\text{r})\,[\text{CLS}] \\
\text{response}(q, C) &= \underset{r \in C}{\text{argmax}}\, h_q \cdot h_r
\end{aligned}
$$

Using information retrieval to grab a response from some corpus that is appropriate given the dialogue context

- think of the **user's turn** as a **query q,**
- **Goal:** retrieve and repeat some appropriate **turn r** as the response from a corpus of **conversations C**



Using a language model or encoder-decoder to generate the response given the dialogue Context.

Using a language model or encoder-decoder to generate the response given the dialogue Context.

- think of the **user's turn** as a **query q,**
- **Goal: generate** each token of the **response** by conditioning on the encoding of the entire **query q** and the response so far.

Using a language model or encoder-decoder to generate the response given the dialogue Context.

- think of the **user's turn** as a **query q,**
- **Goal: generate** each token of the **response** by conditioning on the encoding of the entire **query q** and the response so far.

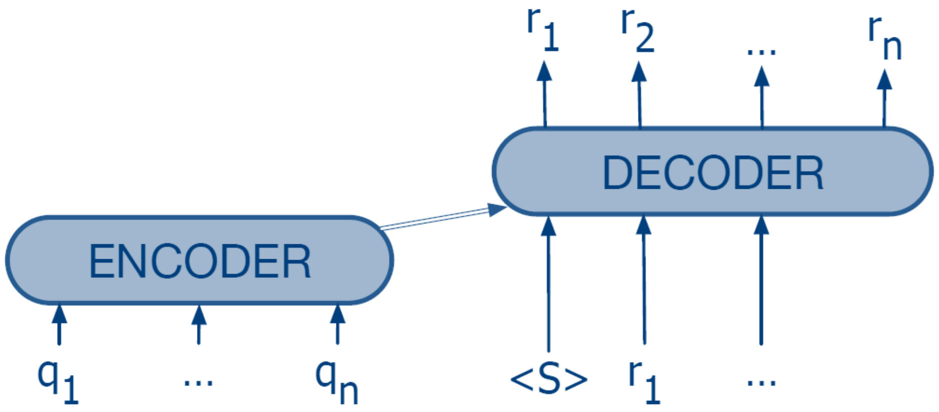
$$\hat{r}_t = \mathrm{argmax}_{\mathrm{w} \in \mathrm{V}} P(w|q, r_1 \ldots r_{t-1})$$

Chatbot/Dialog Systems

Generation-based Response Generation

Using a language model or encoder-decoder to generate the response given the dialogue Context.

- think of the **user's turn** as a **query q,**
- **Goal: generate** each token of the **response** by conditioning on the encoding of the entire **query q** and the response so far.



Automatic Speech Recognition

Introduction

The technology that allows human beings to use their voices to speak with a computer interface in a way that, in its most sophisticated variations, resembles normal human conversation.

A speech recognizer computes the most likely words sequence given a sequence of speech features. For this, speech features are captured and evaluated using an acoustic and a language model.

$$\underbrace{\boldsymbol{w}^*}_{\text{Transcribed Speech}} = ?$$

A speech recognizer computes the most likely words sequence given a sequence of speech features. For this, speech features are captured and evaluated using an acoustic and a language model.

$$\boldsymbol{w}^* = \underset{?}{\operatorname{argmax}}\, P(\;?\;)$$

Transcribed Speech

A speech recognizer computes the most likely words sequence given a sequence of speech features. For this, speech features are captured and evaluated using an acoustic and a language model.

$$\boldsymbol{w}^* = \underset{\boldsymbol{w}}{\operatorname{argmax}}\, P(\boldsymbol{w} \mid ?)$$

Transcribed Speech

A speech recognizer computes the most likely words sequence given a sequence of speech features. For this, speech features are captured and evaluated using an acoustic and a language model.

$$\boldsymbol{w}^* = \operatorname*{argmax}_{\boldsymbol{w}} P(\boldsymbol{w} \mid ?)$$

Transcribed Speech

All possible Word Sequences

A speech recognizer computes the most likely words sequence given a sequence of speech features. For this, speech features are captured and evaluated using an acoustic and a language model.

$$\boldsymbol{w}^* = \underset{\boldsymbol{w}}{\operatorname{argmax}} P(\boldsymbol{w}|\boldsymbol{X})$$

Transcribed Speech

All possible Word Sequences

Sequence of audio features

A speech recognizer computes the most likely words sequence given a sequence of speech features. For this, speech features are captured and evaluated using an acoustic and a language model.

$$\underbrace{\boldsymbol{w}^*}_{\text{Transcribed Speech}} = \underbrace{\underset{\boldsymbol{w}}{\operatorname{argmax}} P(\boldsymbol{w}}_{\text{All possible Word Sequences}} | \underbrace{\boldsymbol{X}}_{\text{Sequence of audio features}}) = ?$$

A speech recognizer computes the most likely words sequence given a sequence of speech features. For this, speech features are captured and evaluated using an acoustic and a language model.

$$\underbrace{\boldsymbol{w}^*}_{\text{Transcribed Speech}} = \underbrace{\operatorname*{argmax}_{\boldsymbol{w}} P(\boldsymbol{w}}_{\text{All possible Word Sequences}} | \underbrace{\boldsymbol{X}}_{\text{Sequence of audio features}}) = \operatorname*{argmax}_{\boldsymbol{w}} \frac{P(\boldsymbol{w}) \cdot P(\boldsymbol{X}|\boldsymbol{w})}{P(\boldsymbol{X})} = ?$$

A speech recognizer computes the most likely words sequence given a sequence of speech features. For this, speech features are captured and evaluated using an acoustic and a language model.

$$\boldsymbol{w}^* = \underset{\boldsymbol{w}}{\operatorname{argmax}}\, P(\boldsymbol{w}|\boldsymbol{X}) = \underset{\boldsymbol{w}}{\operatorname{argmax}} \frac{P(\boldsymbol{w}) \cdot P(\boldsymbol{X}|\boldsymbol{w})}{P(\boldsymbol{X})} = \underset{\boldsymbol{w}}{\operatorname{argmax}}\, P(\boldsymbol{w}) \cdot P(\boldsymbol{X}|\boldsymbol{w})$$

Transcribed Speech

All possible Word Sequences

Sequence of audio features

Not required for speech recognition except a “confidence measure” is needed

A speech recognizer computes the most likely words sequence given a sequence of speech features. For this, speech features are captured and evaluated using an acoustic and a language model.

$$\boldsymbol{w}^* = \underset{\boldsymbol{w}}{\operatorname{argmax}}\, P(\boldsymbol{w}|\boldsymbol{X}) = \underset{\boldsymbol{w}}{\operatorname{argmax}} \frac{P(\boldsymbol{w}) \cdot P(\boldsymbol{X}|\boldsymbol{w})}{P(\boldsymbol{X})} = \underset{\boldsymbol{w}}{\operatorname{argmax}}\, P(\boldsymbol{w}) \cdot P(\boldsymbol{X}|\boldsymbol{w})$$

Transcribed Speech

All possible Word Sequences

Sequence of audio features

Not required for speech recognition except a “confidence measure” is needed

Language Model

$$P(\boldsymbol{w}) = P(w_1) \cdot P(w_2|w_1) \cdot P(w_3|w_1 w_2) \cdot \prod_{i=4}^{m} P(w_i|w_1 \ldots w_{i-1})$$

A speech recognizer computes the most likely words sequence given a sequence of speech features. For this, speech features are captured and evaluated using an acoustic and a language model.

$$\underbrace{\boldsymbol{w}^*}_{\text{Transcribed Speech}} = \underbrace{\operatorname*{argmax}_{\boldsymbol{w}} P(\boldsymbol{w}}_{\text{All possible Word Sequences}}|\underbrace{\boldsymbol{X}}_{\text{Sequence of audio features}}) = \operatorname*{argmax}_{\boldsymbol{w}} \frac{P(\boldsymbol{w}) \cdot P(\boldsymbol{X}|\boldsymbol{w})}{\underbrace{P(\boldsymbol{X})}_{\text{Not required for speech recognition except a “confidence measure” is needed}}} = \operatorname*{argmax}_{\boldsymbol{w}} \underbrace{P(\boldsymbol{w})}_{\text{Language Model}} \cdot \underbrace{P(\boldsymbol{X}|\boldsymbol{w})}_{\text{Acoustic Model}}$$

Automatic Speech Recognition, is the technology that allows human beings to use their voices to speak with a computer interface in a way that, in its most sophisticated variations, resembles normal human conversation.

Automatic Speech Recognition

Introduction

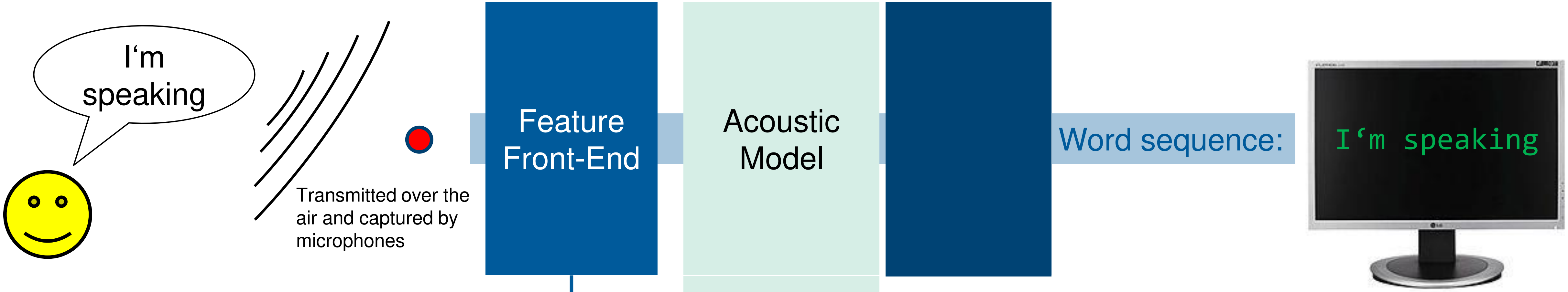
Automatic Speech Recognition, is the technology that allows human beings to use their voices to speak with a computer interface in a way that, in its most sophisticated variations, resembles normal human conversation.

$$\underline{\hat{w}} = \operatorname{argmax} P(\underline{x} | \underline{w}) P(\underline{w})$$

Automatic Speech Recognition, is the technology that allows human beings to use their voices to speak with a computer interface in a way that, in its most sophisticated variations, resembles normal human conversation.

$$\underline{\hat{w}} = \operatorname{argmax} P(\underline{x}|\underline{w})P(\underline{w})$$

Automatic Speech Recognition, is the technology that allows human beings to use their voices to speak with a computer interface in a way that, in its most sophisticated variations, resembles normal human conversation.

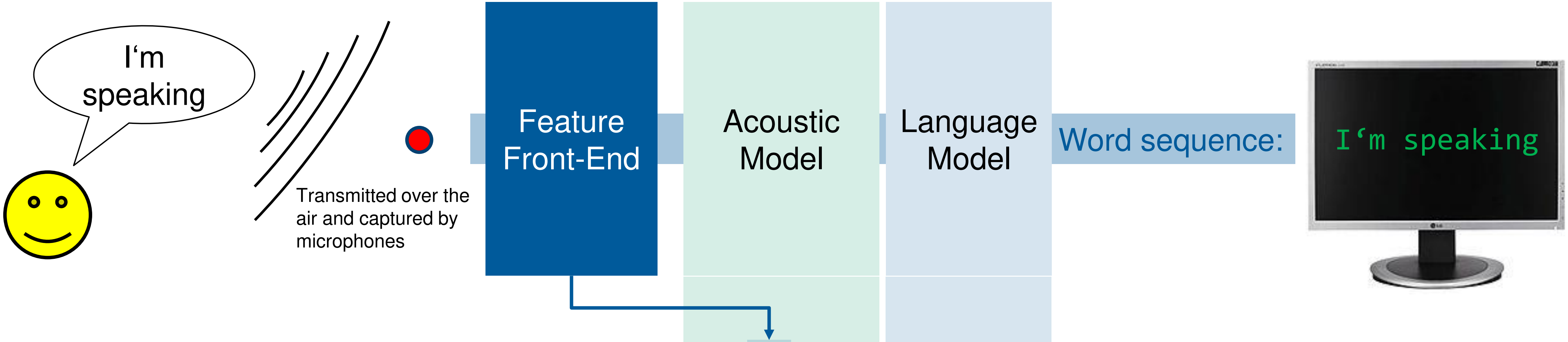


$$\underline{\hat{w}} = \operatorname{argmax} P(\underline{x}|\underline{w})P(\underline{w})$$

Automatic Speech Recognition

Introduction

Automatic Speech Recognition, is the technology that allows human beings to use their voices to speak with a computer interface in a way that, in its most sophisticated variations, resembles normal human conversation.

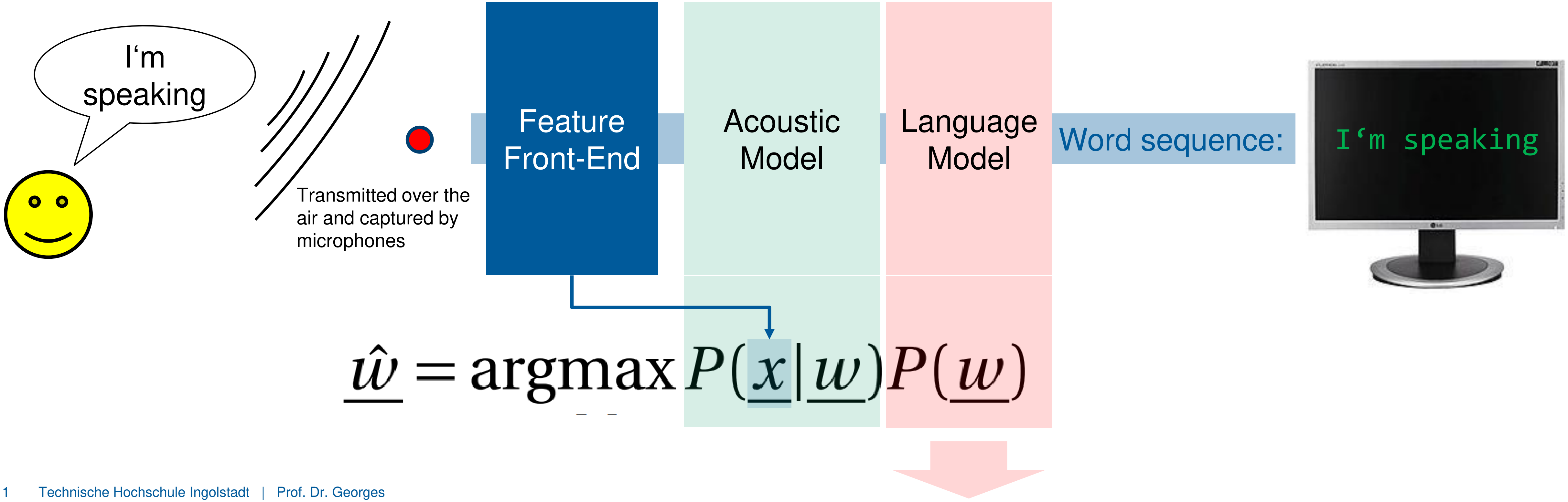


$$\hat{w} = \operatorname{argmax} P(x|w)P(w)$$

Automatic Speech Recognition

Introduction

Automatic Speech Recognition, is the technology that allows human beings to use their voices to speak with a computer interface in a way that, in its most sophisticated variations, resembles normal human conversation.



$$\hat{w} = \operatorname{argmax} P(x|w)P(w)$$

Statistical Language Model

N-gram SLM

A probability distribution over sequences of words. The language model provides context to distinguish between words and phrases that sound similar. Data sparsity is a major problem in building language models. Most possible word sequences are not observed in training. One solution is to make the assumption that the probability of a word only depends on the previous n words. This is known as an n-gram model.

$$P(\underline{w}) = ?$$

A probability distribution over sequences of words. The language model provides context to distinguish between words and phrases that sound similar. Data sparsity is a major problem in building language models. Most possible word sequences are not observed in training. One solution is to make the assumption that the probability of a word only depends on the previous n words. This is known as an n-gram model.

$$P(\underline{w}) = P(w_0 w_1 \ldots w_n w_{n+1})$$

$$= ?$$

Statistical Language Model

N-gram SLM

A probability distribution over sequences of words. The language model provides context to distinguish between words and phrases that sound similar. Data sparsity is a major problem in building language models. Most possible word sequences are not observed in training. One solution is to make the assumption that the probability of a word only depends on the previous n words. This is known as an n-gram model.

$$
\begin{aligned}
P(\underline{w}) &= P(w_0 w_1 \ldots w_n w_{n+1}) \\
&= P(w_0) P(w_1 | w_0) P(w_2 | w_0 w_1) \cdots P(w_n | w_1 w_2 \ldots w_{n-1}) \\
&= \textbf{?}
\end{aligned}
$$

Statistical Language Model

N-gram SLM

A probability distribution over sequences of words. The language model provides context to distinguish between words and phrases that sound similar. Data sparsity is a major problem in building language models. Most possible word sequences are not observed in training. One solution is to make the assumption that the probability of a word only depends on the previous n words. This is known as an n-gram model.

$$
\begin{aligned}
P(\underline{w}) &= P(w_0 w_1 \ldots w_n w_{n+1}) \\
&= P(w_0)P(w_1|w_0)P(w_2|w_0 w_1)\cdots P(w_n|w_1 w_2 \ldots w_{n-1}) \\
&= \prod_{i=1}^{|\underline{w}|} P(w_i|w_1 \ldots w_{i-1}) \\
&\approx \text{?}
\end{aligned}
$$

Statistical Language Model

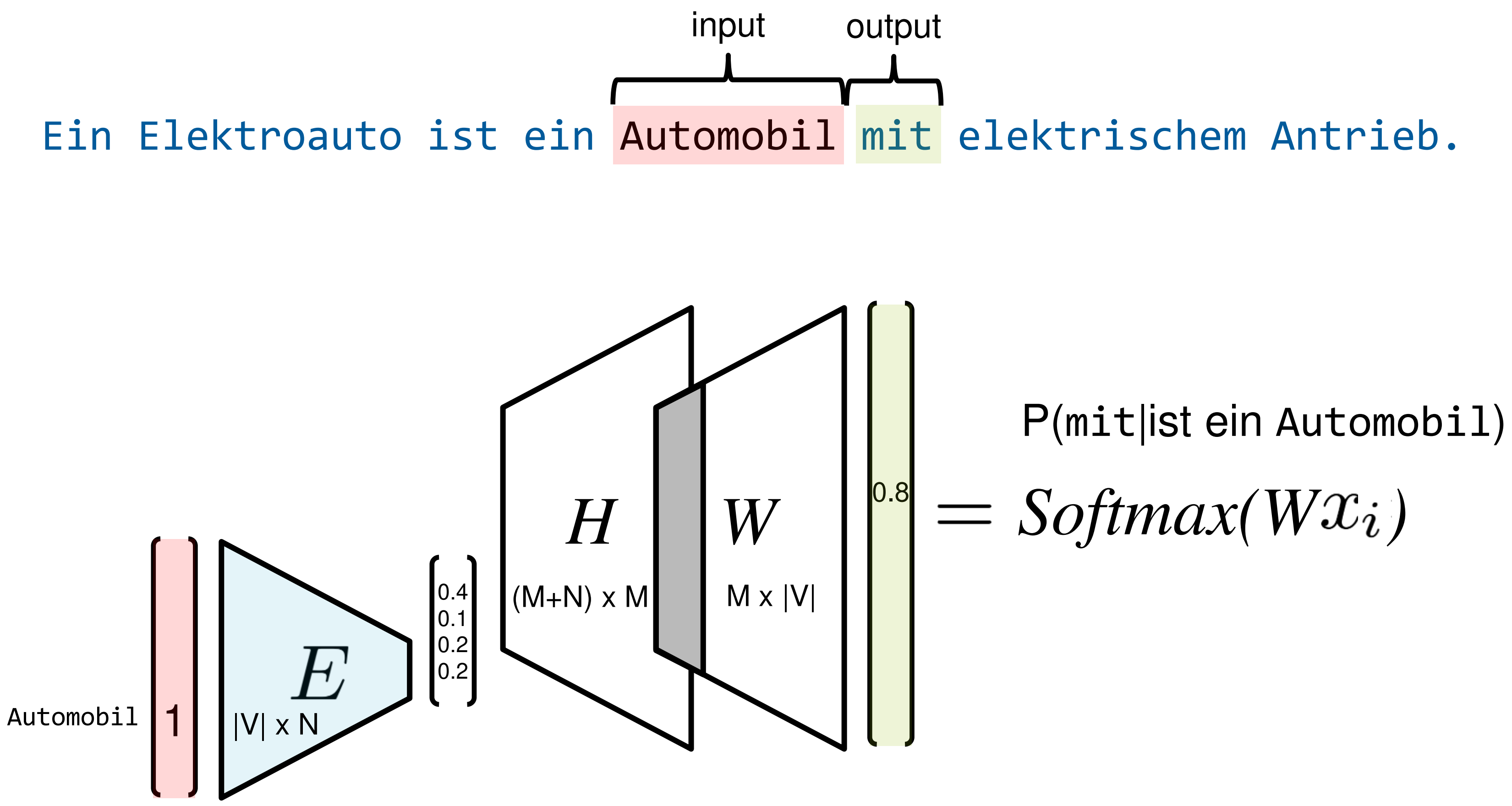
N-gram SLM

A probability distribution over sequences of words. The language model provides context to distinguish between words and phrases that sound similar. Data sparsity is a major problem in building language models. Most possible word sequences are not observed in training. One solution is to make the assumption that the probability of a word only depends on the previous n words. This is known as an n-gram model.

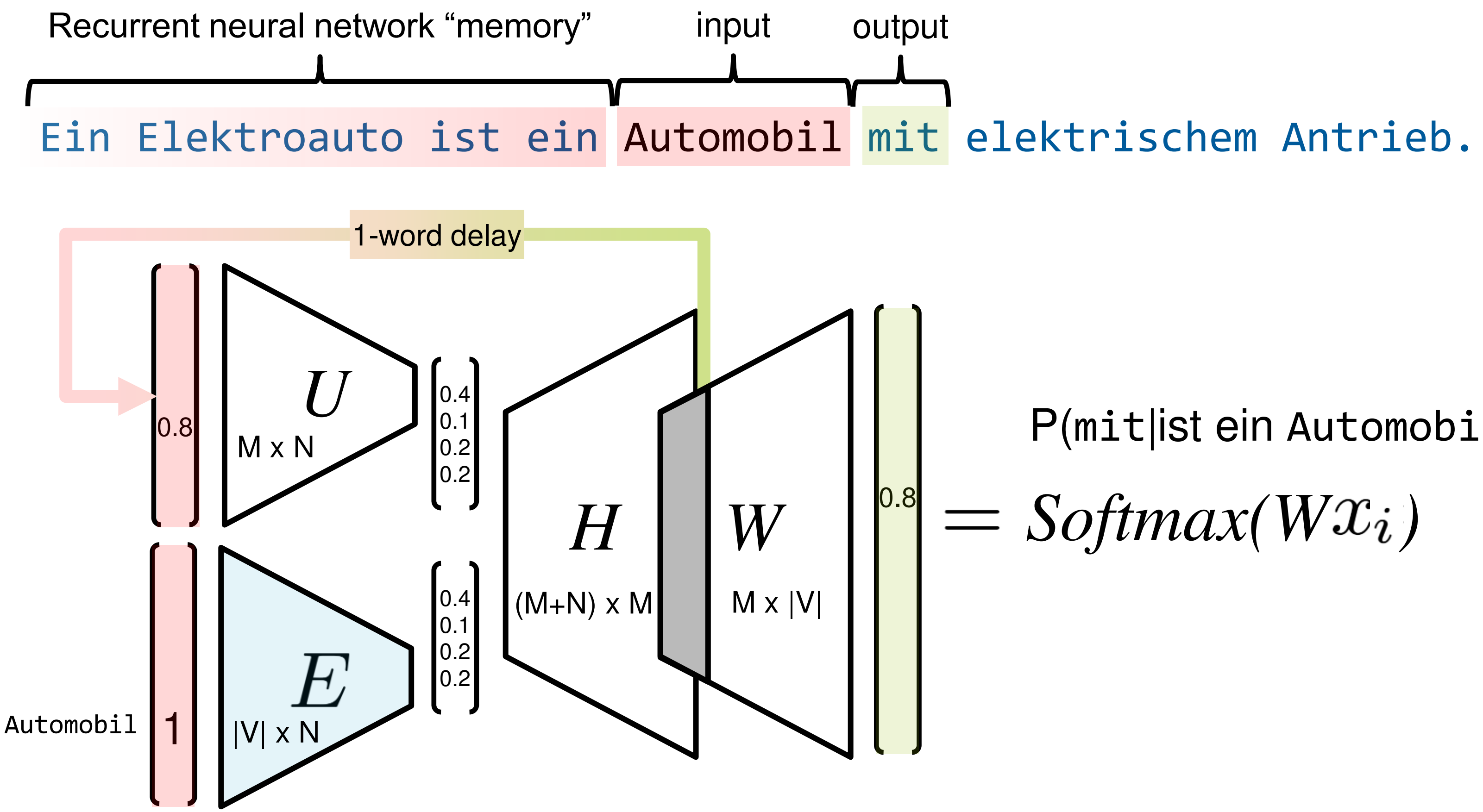
$$
\begin{aligned}
P(\underline{w}) &= P(w_0 w_1 \ldots w_n w_{n+1}) \\
&= P(w_0) P(w_1 | w_0) P(w_2 | w_0 w_1) \cdots P(w_n | w_1 w_2 \ldots w_{n-1}) \\
&= \prod_{i=1}^{|\underline{w}|} P(w_i | w_1 \ldots w_{i-1}) \\
&\approx \prod_{i=1}^{|\underline{w}|} P(w_i | w_{i-n+1} \ldots w_{i-1})
\end{aligned}
$$

$$
P(w_i | w_{i-1}) = \frac{P(w_i w_{i-1})}{P(w_{i-1})} = \frac{\frac{|w_i w_{i-1}|}{|corpus|}}{\frac{|w_{i-1}|}{|corpus|}} = \frac{|w_i w_{i-1}|}{|w_{i-1}|}
$$

Neural Language Model: How likely is the next word?



Neural Language Model: How likely is the next word?



P(mit|ist ein Automobil)

$$= Softmax(Wx_i)$$