Technische Hochschule Rosenheim

Applications of & Introduction to Artificial Intelligence (A2I2)

Applications of
Language Understanding and Speech Recognition

Technische Hochschule Rosenheim

Sommer 2020

Prof. Dr. M. Tilly

Agenda

- Applications of Natural Language Processing
- Intent detection with Language Understanding Service
- Speech Recognition
- Speech Translation

Why NLP?

- The objective of NLP is to make computer/machines as *intelligent* as human beings in understanding language.
- The ultimate goal of NLP is to the fill the gap how people communicate (natural language) and what the computer understands (machine language).
- NLP can be used to organize and structure knowledge to perform tasks such as
  - Automatic text summarization
  - Translation
  - Named Entity Recognition
  - relationship extraction
  - Sentiment Analysis
  - Speech Recognition
  - Topic segmentation

Traditional definition of NLP: the branch of AI

- Deal with analyzing, understanding and generating the languages that humans use naturally (natural language)

- Study knowledge of language at different levels
  - Phonetics and Phonology – the study of linguistic sounds
  - Morphology – the study of the meaning of components of words
  - Syntax – the study of the structural relationships between words
  - Semantics – the study of meaning
  - Discourse – they study of linguistic units larger than a single utterance

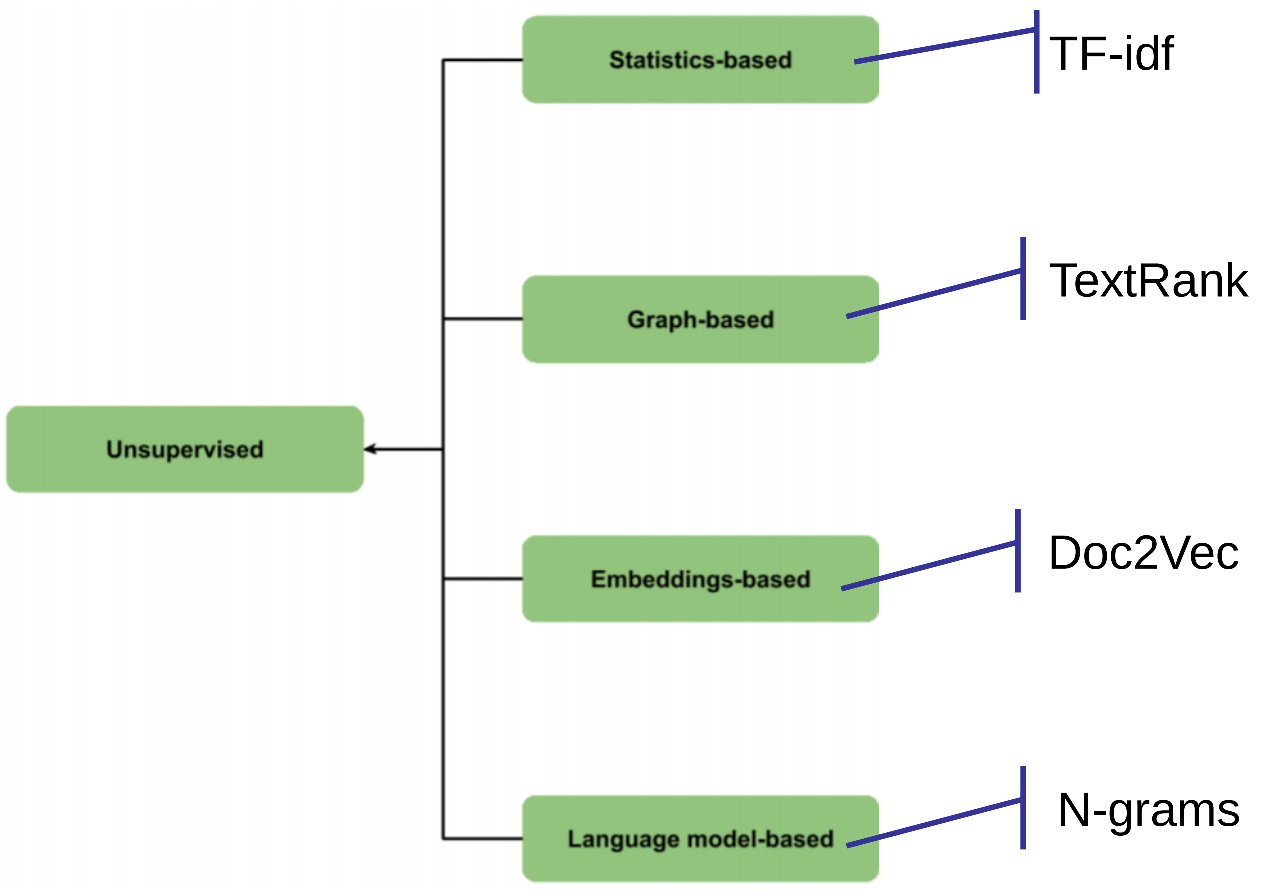
Pragmatic definition: building computer systems

- Process large text corpora -> turning information into knowledge
  - Text classification
  - Information retrieval and extraction
  - Machine reading comprehension and question answering (QNA)

- Enable human-computer interactions, making knowledge accessible to humans in the most natural way
  - Dialogue and conversational agents
  - Machine translation

Unsupervised Key Phrase detection

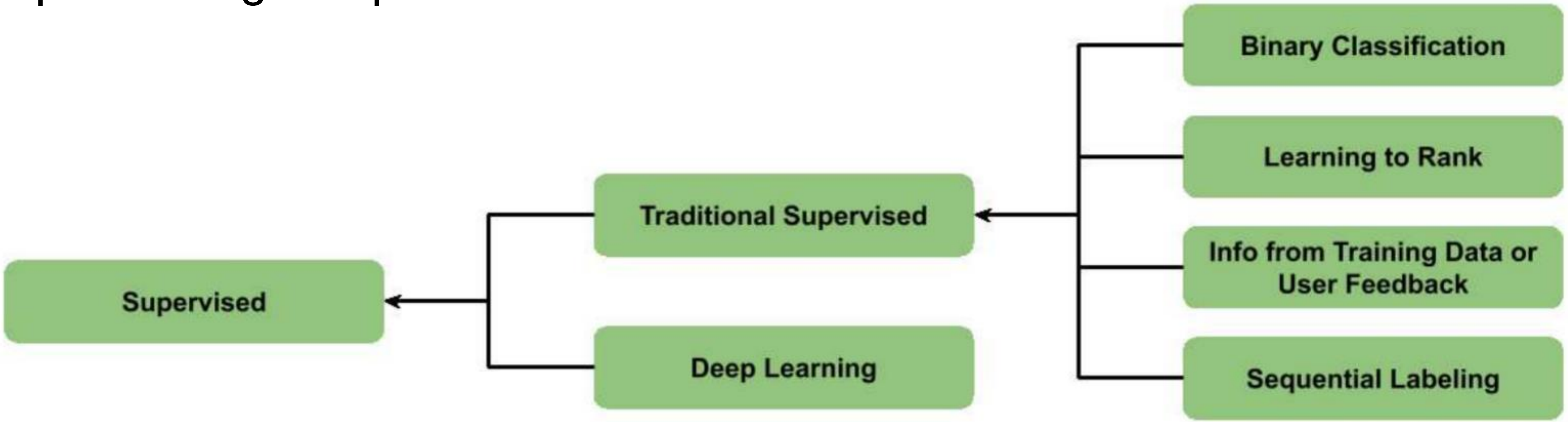
- Selection of the candidate lexical units based on some heuristics
  - Examples of such heuristics are the exclusion of stopwords and the selection of words that belong to a specific part-of-speech (POS).
- Ranking of the candidate lexical units
- Formation of the *keyphrases* by selecting words from the top-ranked ones or by selecting a phrase with a high rank score or whose parts have a high score



taken from 'A Review of Keyphrase Extraction', Eirini Papagiannopoulou and Grigorios Tsoumakas, https://arxiv.org/pdf/1905.05044.pdf

Supervised Methods

- A classifier is trained on annotated with *keyphrases* documents in order to deter
- Learning to rank approaches learn a ranking function that sorts the candidate phrases based on their score of being a keyphraseine whether a candidate phrase is a *keyphrase* or not.
- Deep Learning: Sequence Model with RNNs



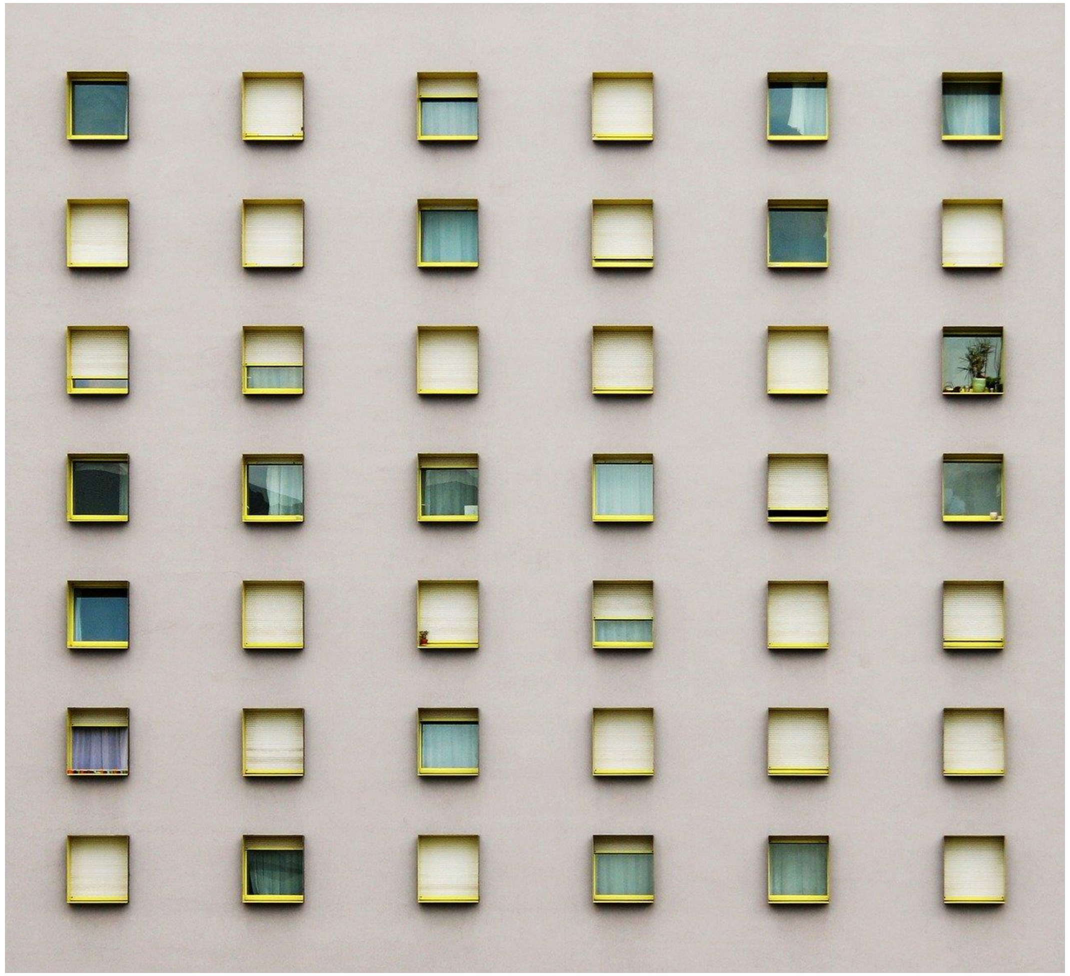
taken from ‘A Review of Keyphrase Extraction’, Eirini Papagiannopoulou and Grigorios Tsoumakas, https://arxiv.org/pdf/1905.05044.pdf

Sequence Model

- Sequence modeling: "*This morning I took the dog for a* ______."
  - Fixed window could use for example last 2 words for prediction!
- Long-term dependencies:
  - "In France, I had a great time and I learnt some of the ______ language."
- We need information from the past (and maybe future) to guess.
  - Use entire sequence as a set of count: Bag of words!
  - But "*The food was good, not bad at all*." vs. "*The food was bad, not good at all*."
- References as parameter: "I like the book. Normally I do not like thriller but this one is great".

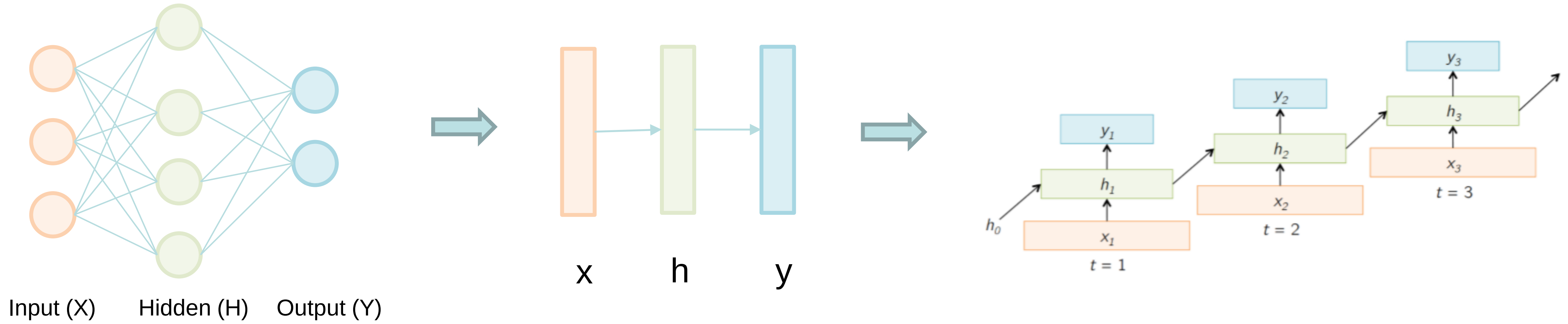
Sequence Models

- Sequence models are great if we need
  - to deal with variable-length in sequences
  - to maintain sequence order
  - to keep track of long-term dependencies
  - to share parameters across the sequence



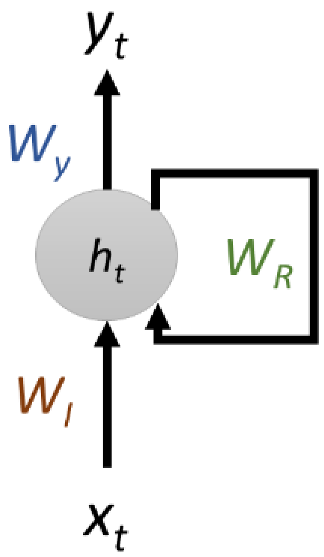
Recurrent Neural Networks

- Recurrent neural networks (RNNs) are connected models with the ability to selectively pass information across sequence steps, while processing sequential data one element at a time.
- Recurrent Neural Networks
  - take the previous output or hidden states as inputs
  - The composite input at time *t* has some historical information about the happenings at time T < t



Vanilla RNN Cell

- Recurrent neural networks (RNNs) are connected models with the ability to selectively pass information across sequence steps, while processing sequential data one element at a time.



$$h_t = g_h(W_I x_t + W_R h_{t-1} + b_h)$$

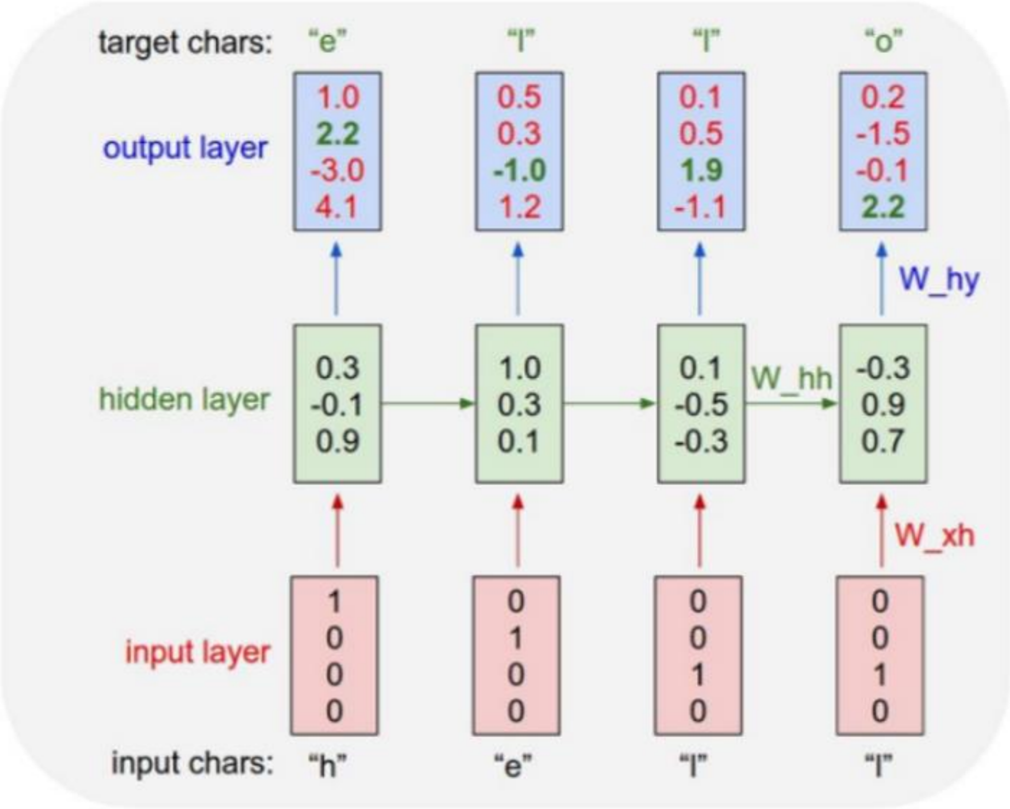
$$y_t = g_y(W_y h_t + b_y)$$

A Sequence Model

Example: Character-level Language Model

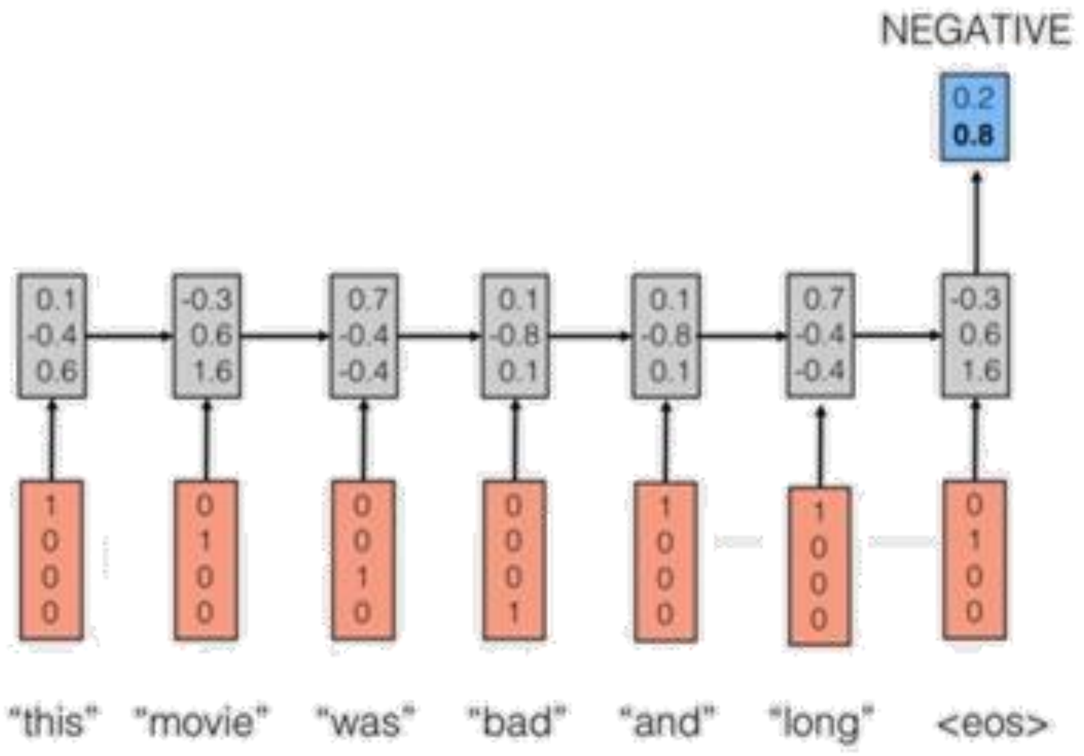
Vocabulary: [h,e,l,o]

Example training sequence: **“hello”**



Sentiment Analysis

- Positive or negative movie review?
  - unbelievably disappointing
  - Full of zany characters and richly applied satire, and some great plot twists
  - this is the greatest screwball comedy ever filmed
  - It was pathetic. The worst part about it was the boxing scenes.



Applications

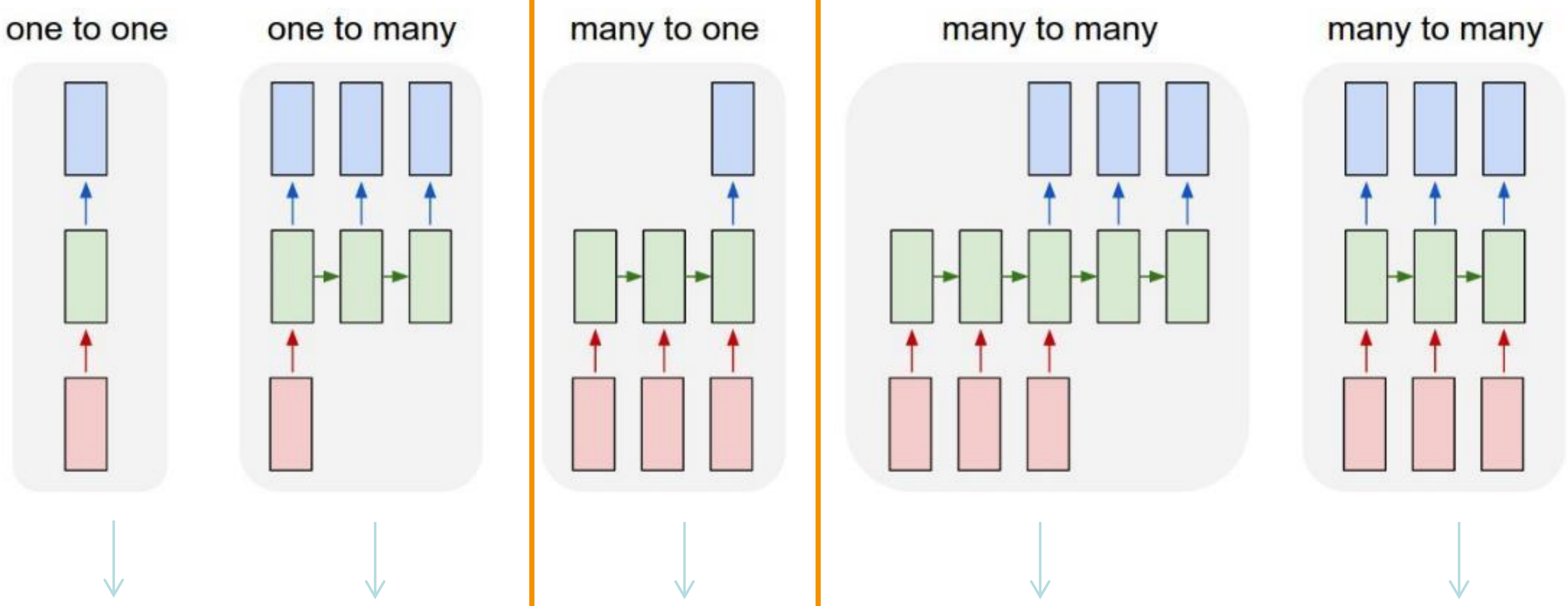


Image classification

Image captioning

Sentiment analysis

Machine translation

Video classification

Modern Applications

- Search Engines
  - Google, Bing, Baidu, etc.
- Question Answering
  - IBM's Watson, QnA Maker, …
- Natural Language Assistants
  - Amazon Alexa, Apple's Siri, OK Google,
- Translation Systems
  - Google Translate
- Chatbots
  - Google Flow
- Text Analytics, …

Text Analytics *as a Service*

- Extract insights from text
  - Uses unstructured text
  - Identifies the **language**
  - Identifies **key phrases**
  - Identifies **entities** (e.g people, places, and organizations)
    - Named entities
    - Linked entities
  - **Sentiment** detection

- IBM Watson
- Microsoft Azure Text Analytics
- AWS Comprehend
- Google Cloud Natural Language

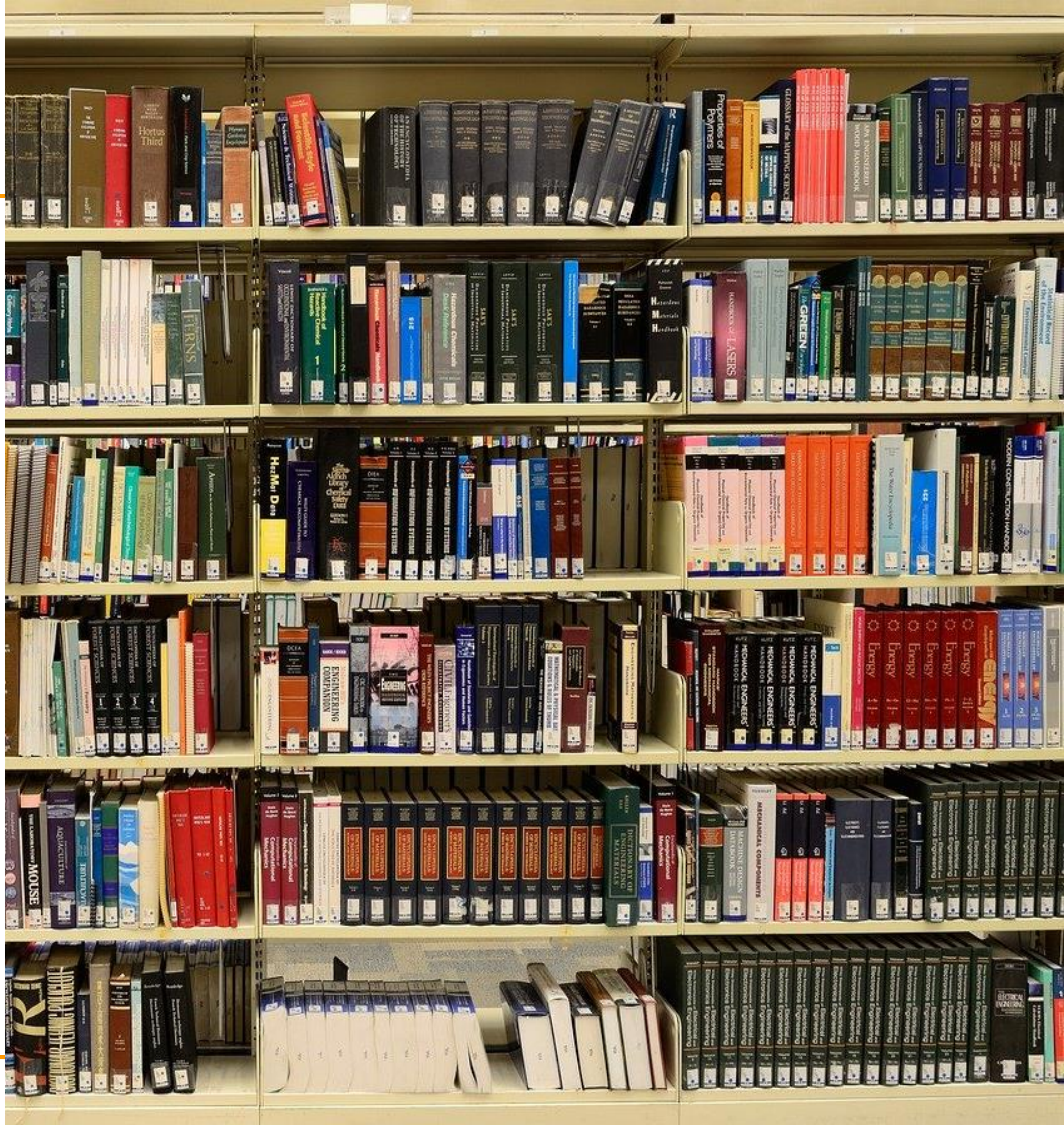
Intent Detection/ Classification

- *Intent classification* is the automated association of text to a specific purpose or goal.
  - A classifier analyzes pieces of text and categorizes them into intents such as *Purchase*, *Downgrade*, *Unsubscribe*, and *Demo Request*
  - Applications are customer queries, emails, chat conversations, social media comments, …
  - It helps to automate processes

- Supervised Learning approach
  - Labeled training data (sentence – intent)
  - ‘what is the forecast for tomorrow?’ – ‘weatherInfo’
  - **Feature(‘*what is the forecast for tomorrow?*’) – Y(‘*weatherInfo*’)**

But…

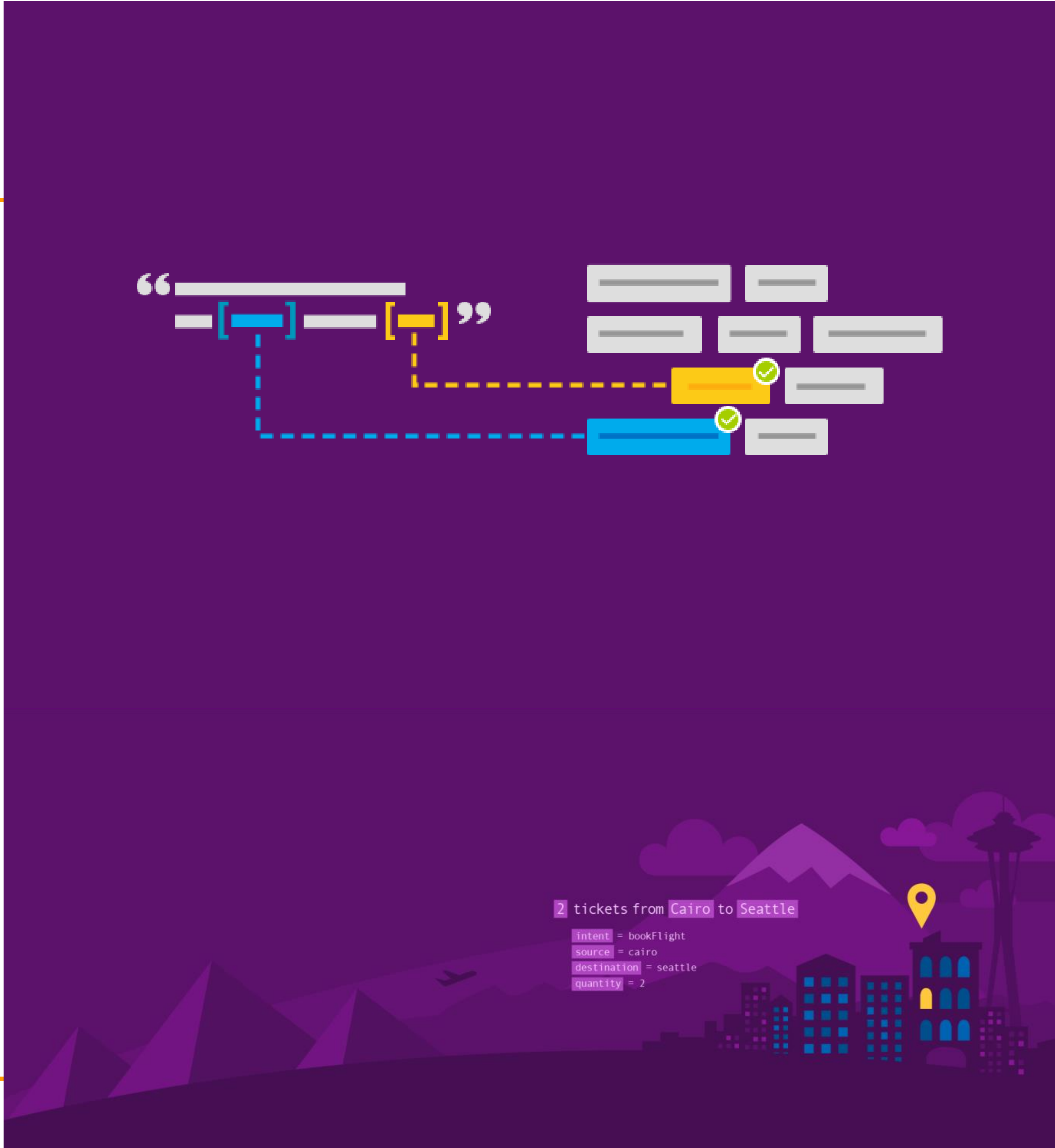
- Which feature(s) should I select?
- Data, data, data, ….
- Training
- Validation



Language Understanding Cognitive Service (LUIS)

- Language Understanding (LUIS) is a cloud-based API service that applies custom machine-learning intelligence to a user's conversational, natural language text to predict overall meaning, and pull out relevant, detailed information.

- A client application for LUIS is any conversational application that communicates with a user in natural language to complete a task. Examples of client applications include social media apps, chat bots, and speech-enabled desktop applications.



The Three Main LUIS Objects

- **Intents** - An intent represents a task or action the user wants to perform.
- **Utterances** - Utterances are input from the user that your app needs to interpret.
- **Entities** - The entity represents a word or phrase inside the utterance that you want extracted.

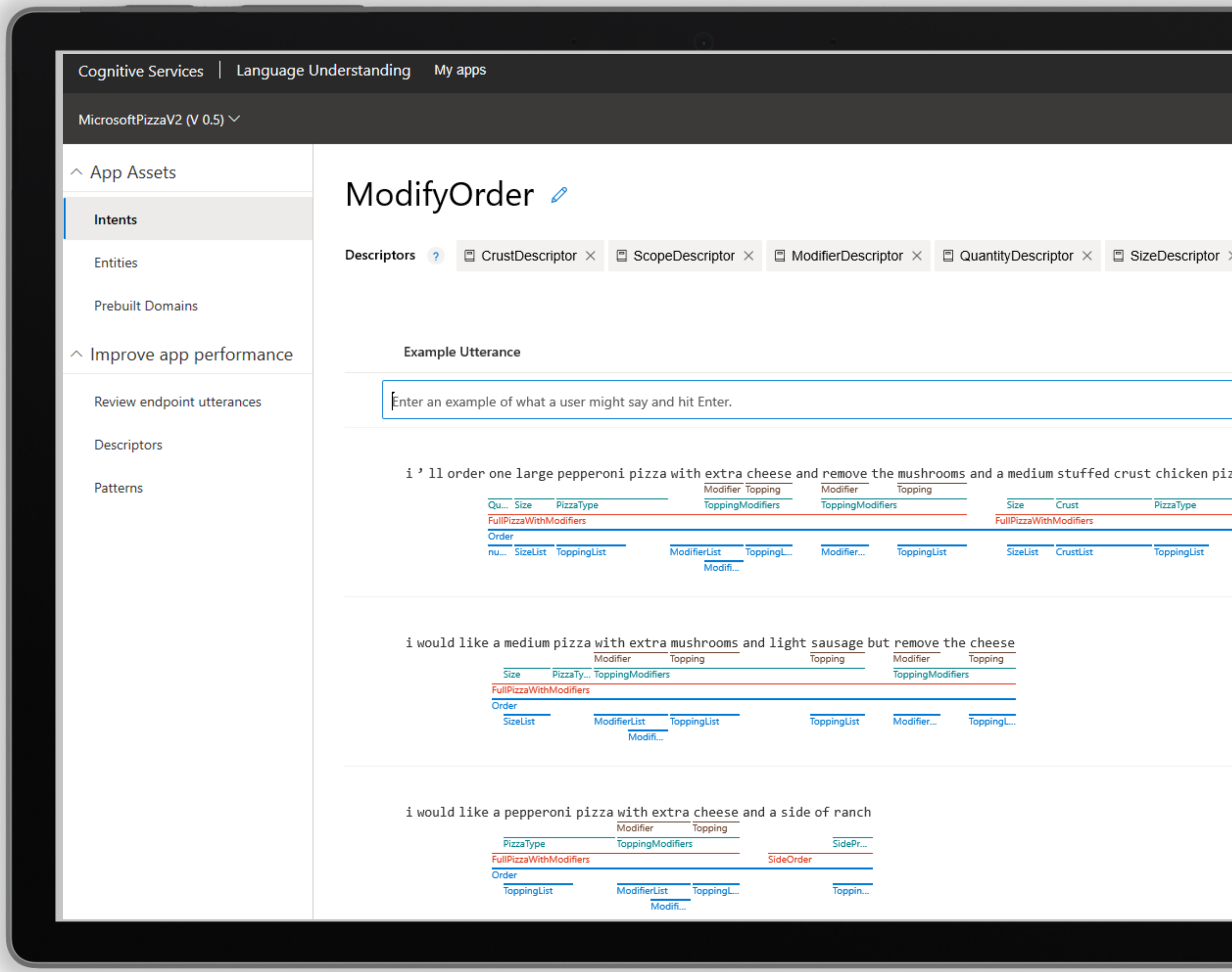
Language Understanding *Intelligent Service* (LUIS)

Core language understanding

- Provide sample utterances and define intents
- Mark entities
- Predict intents

Enhanced developer experience

- Browser
- API for various languages



Example Utterance

How can we create a scheme for this

- Entities include product, size and quantity etc..
- Map actions to intents (such as adding, removing, modifying, canceling etc..)

quantity | Size | Product | Options Attribute | Toppings

I’ll order one large pepperoni pizza with extra cheese

```
"entities":

Simple: "Product“

ClosedLists: "CrustType","CutOptions","Dips", “OrderType”,
"paymentType","PizzaType”,"Sauces","Sizes", "Toppings”, “OptionsAttribute”

PrebuiltEntities: "datetimeV2", "number"
```

```
"intents": [
    {"name": “MakeOrder"},
    {"name": “RemoveOrder"},
    {"name": “EditOrder"},
    {"name": “CancelOrder"},
    {"name": "Confirmation"},
    {"name": "Greetings"},
    {"name": "None"}
  ],
```

Language Understanding (LUIS): Entity Usage

Simple Entity: Should be the entity type to use by default

Roles: When the same entity could be used in different contexts (e.g. number could be quantity “three pizzas” or type “four cheese pizza”).

Composite Entity: Used to encode structure, could be multiple entities, or intent/entity elements (e.g. remove {topping})

List Entity: Suitable for a bounded set of elements that rarely change (e.g. pizza size, payment method, etc…)

Prebuilt Entity: Well formed entities that should be leveraged when needed (e.g. datetime, number etc…)

Regex Entity: Well formed alpha-numeric elements (e.g. product codes)

Prebuilt List Simple Intent List

I’ll order one large pepperoni pizza with extra cheese and remove the mushrooms

Composite Actional Composite

Language Understanding (LUIS): Entity Types

Simple Entity: Describes a single concept. They learn from context using examples through machine learning

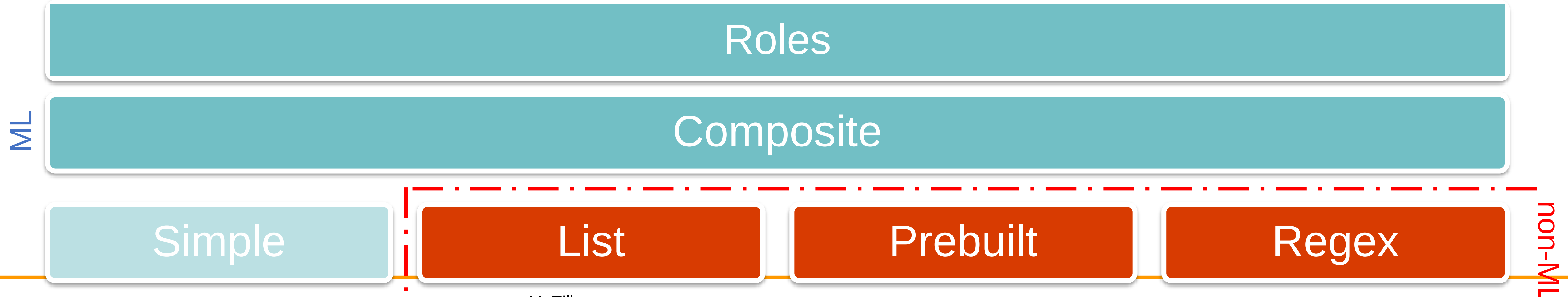
Roles: Named, contextual subtypes of an entity. Extract more information by adding roles to your entities. Roles can be applied to any kind of entity.

Composite Entity: Represents an entity that has parts. It is made up of one or more entities, including supplementary text.

List Entity: A fixed list of words, phrases, and their synonyms that get exact matched in your utterances.

Prebuilt Entity: Common entities provided by LUIS through open source text recognizers available through GitHub, expanded through the community and driven by Microsoft.

Regex Entity: An entity that exact matches based on the regular expression defined.



Bottom-Up Entity Decomposition

Technische Hochschule Rosenheim

Actionable composites ("Int-ent-ities")

- Map actions to composite entities (such as adding, removing, modifying, canceling etc..)

number | Size | Product | Options Attribute | Toppings | Options Qualifier | Toppings | Size | CrustType | Product | Dips

I'll order one large pepperoni pizza with extra cheese and remove the mushrooms and a medium stuffed crust chicken pizza with ranch

Full Product | Qualified Option | Remove Item | Full Product

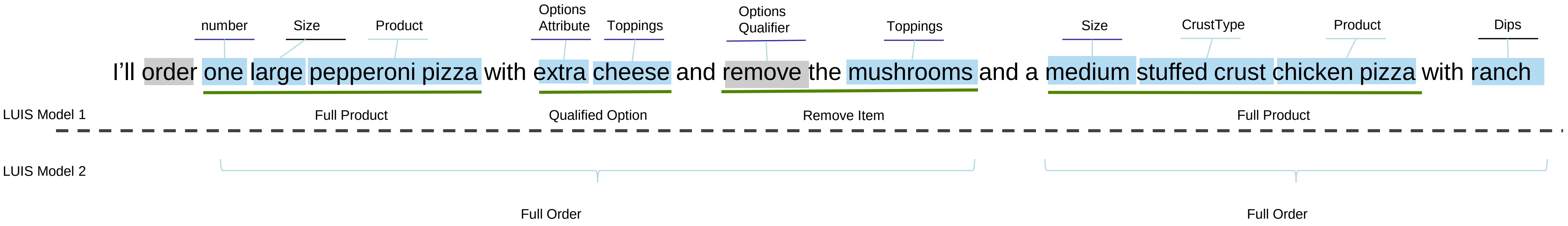
```
"intents“:

    {"name": “ModifyOrder"},
    {"name": “CancelOrder"},
    {"name": "Confirmation"},
    {"name": "Greetings"},
    {"name": "None"}
```

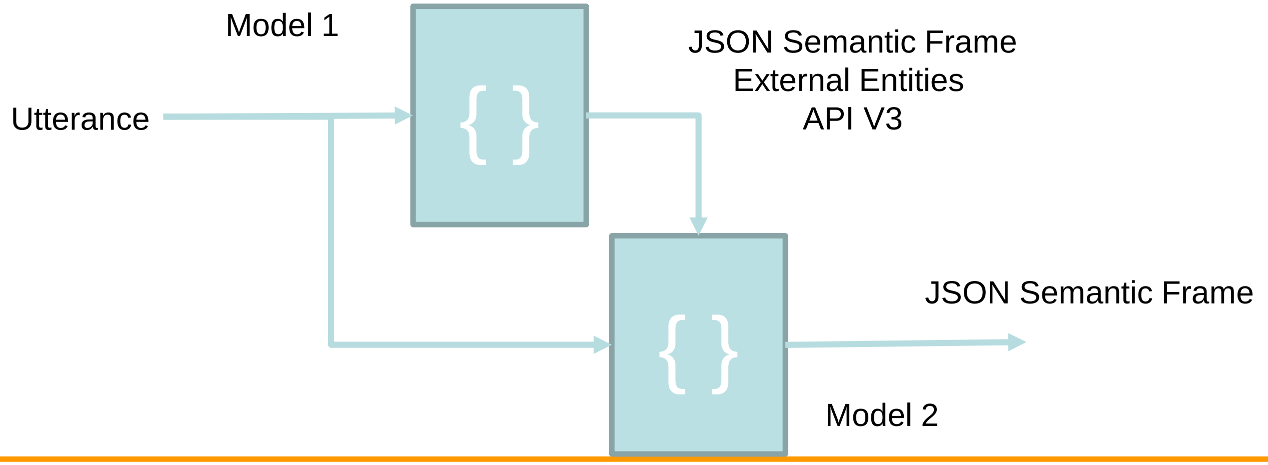
```
"entities":

Simple: "Product“

Composites: "FullProduct", “QualifiedDelivery", "QualifiedOptions", "RemoveItem",
"ReplaceItem", "SideOrder”

ClosedLists: "CrustType","CutOptions","Dips","ModifyOrderDecor", "OPtionsQualifier”,
“OrderType”,"paymentType","PizzaType”,"Sauces","Sizes", "Toppings

PrebuiltEntities: "datetimeV2", "number"
```

Bottom-Up Entity Decomposition

Multi-level Parsing

- Use staged hierarchy to perform the multi-level parse of an utterance





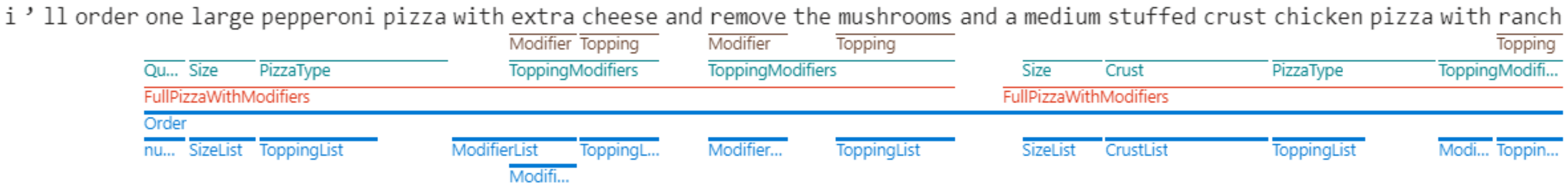
Able to capture the elements of the utterance

- Determine the building elements of the order

Using the external entities and API V3 to provide the elements as input to the next level model

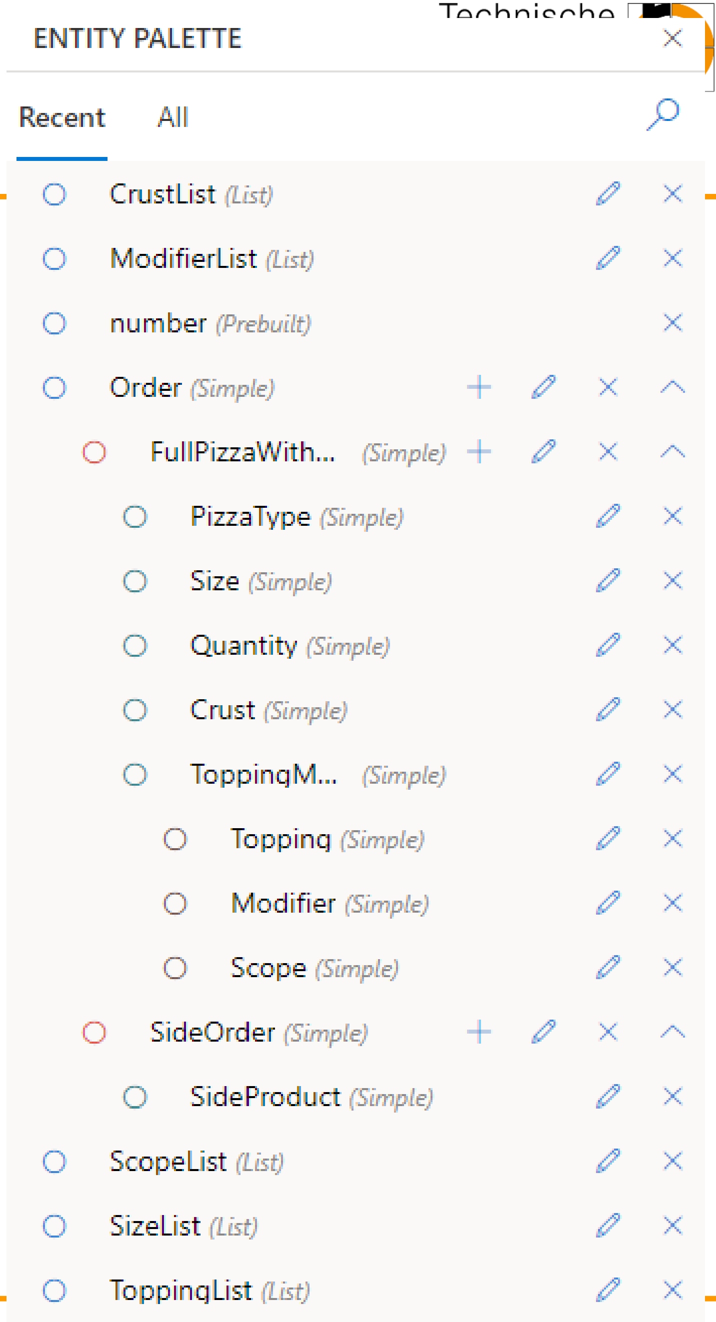
Top-Down Entity Decomposition

Increase the depth of the hierarchy that could be defined



Allow better decomposition of entities

- Improve the ability to conceptualize the decomposition
- Reduce the amount of rework required with the addition of scheme edits
- Allow to build more sophisticated solutions with less effort



Languages/Pre-built models

Supporting more languages for pre-built entities

Full support (All Pre-built entities supported)	Major Support (All pre-built entities except Person & Geo.)	Partial Support (Some pre-built entities supported, Person & Geo not supported)
English (US)	Chinese, French (FR,CA), German, Italian, Portuguese (BR), Spanish (ES, MX), Turkish	Dutch, Japanese, Korean, **Hindi, Gujarati, Telugu, Arabic**

New Pre-built domains

English (US), Chinese, Dutch, French (FR), German, Italian, Japanese, Korean, Portuguese (BR), Spanish (ES), Turkish

More coverage upcoming

- Calendar
- Communication
- Email
- Note
- Places
- Restaurant
- Weather
- To Do
- Utilities
- Web
- Home Automation
- **More to come**

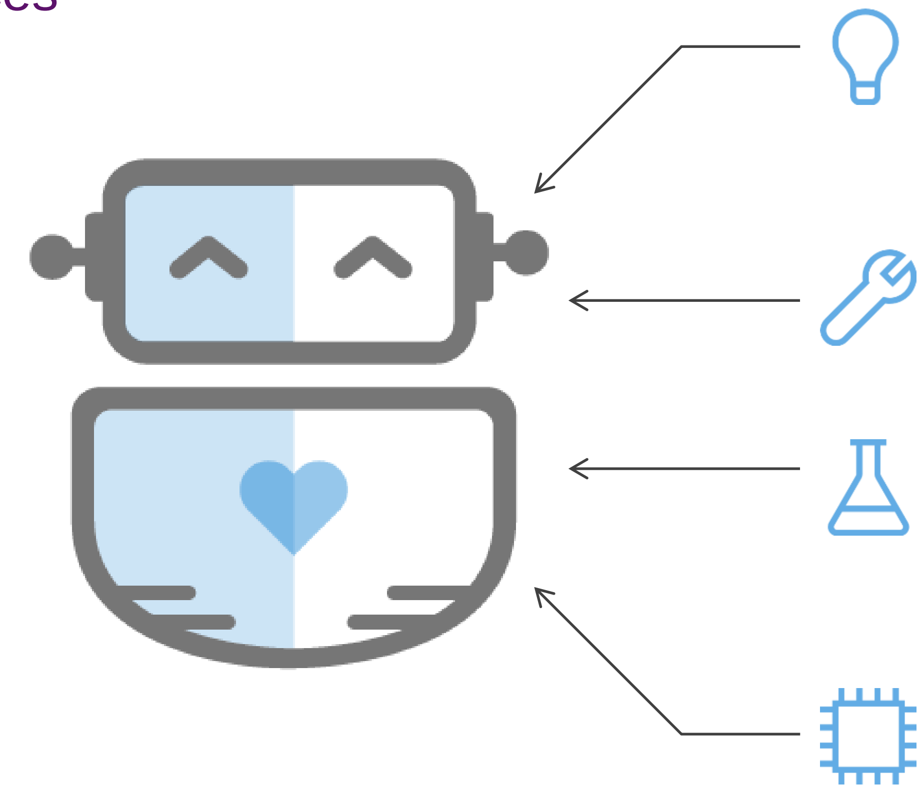
Building a Custom Virtual Assistant: Language Understanding & Bot Framework Skills

Technische Hochschule Rosenheim

Bot Framework: Building conversational apps & experiences

Bot Framework Skills:

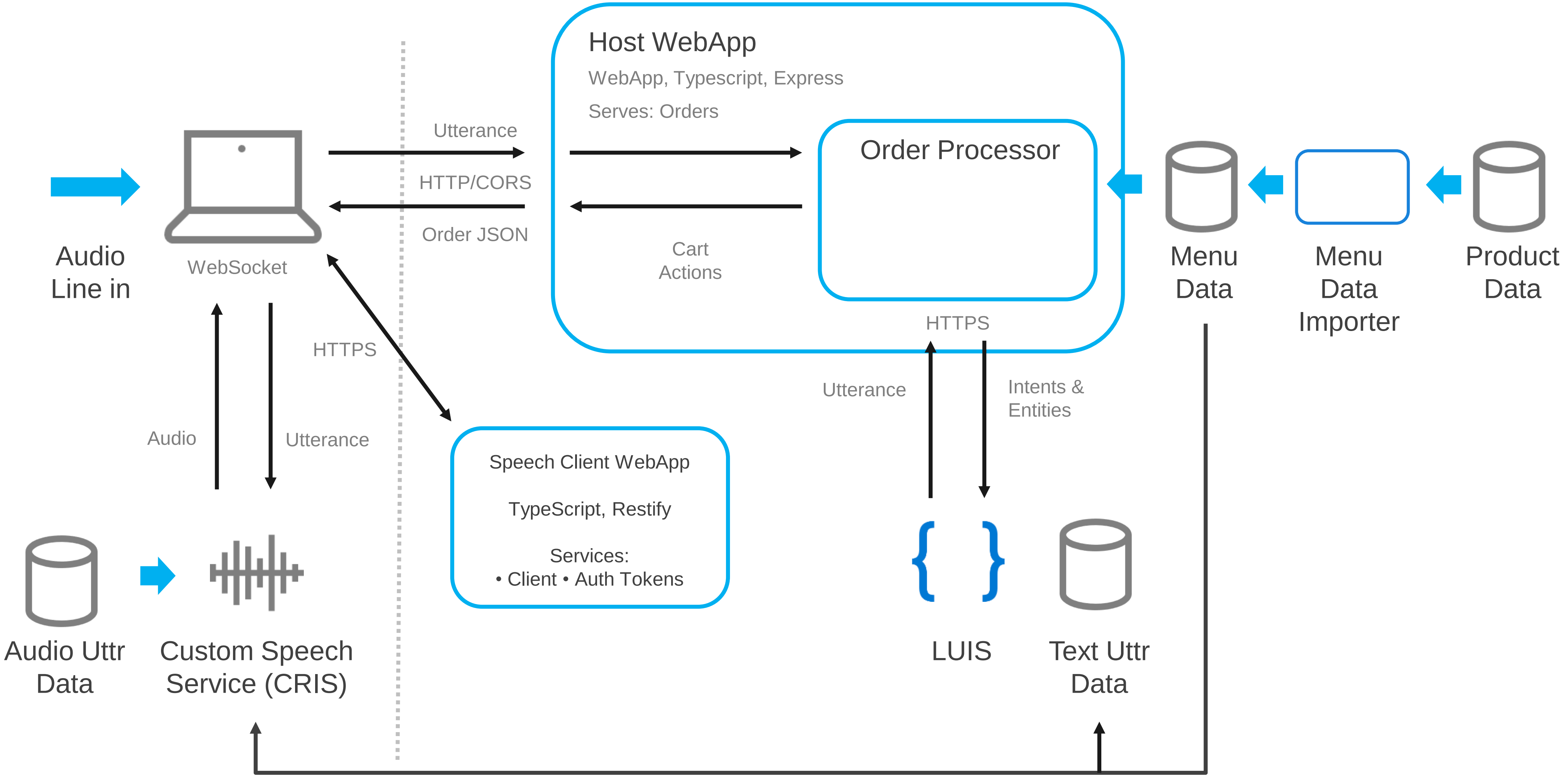
- Uses LUIS & Cognitive Services (QnA, Speech, etc.)
- Include pre-built domains/language models, dialogs, and adaptive cards (GUI)
- Dispatch into prebuilt and custom skills
- Skills contains a self-describing Skill Manifest
- Integrated analytics



Skills are open source, fully customizable

- Skill Template
- C# and Typescript support

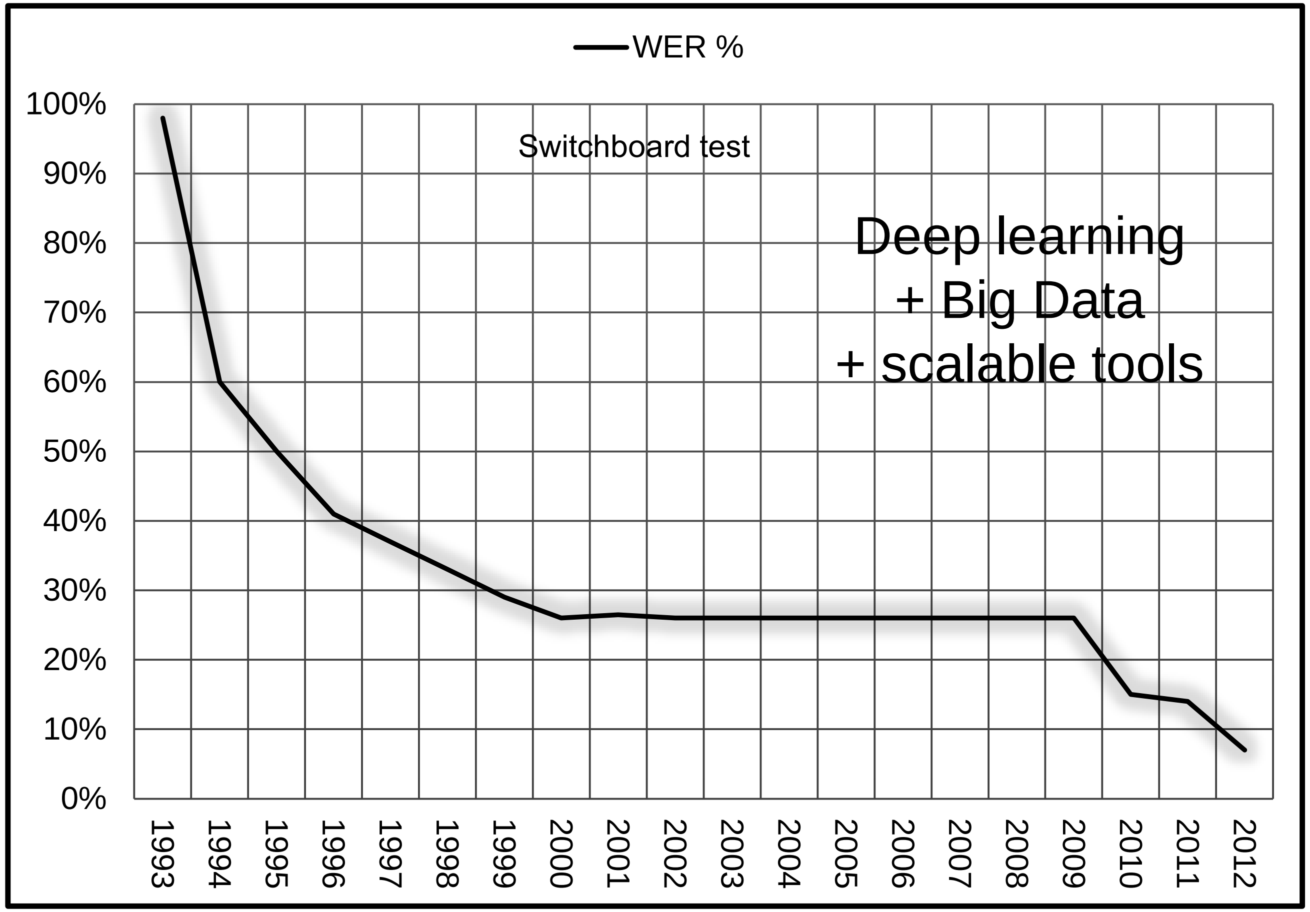
Pizza Ordering Architecture



Once upon in a time …

Rich Rashid in Tianjin, October, 25, 2012

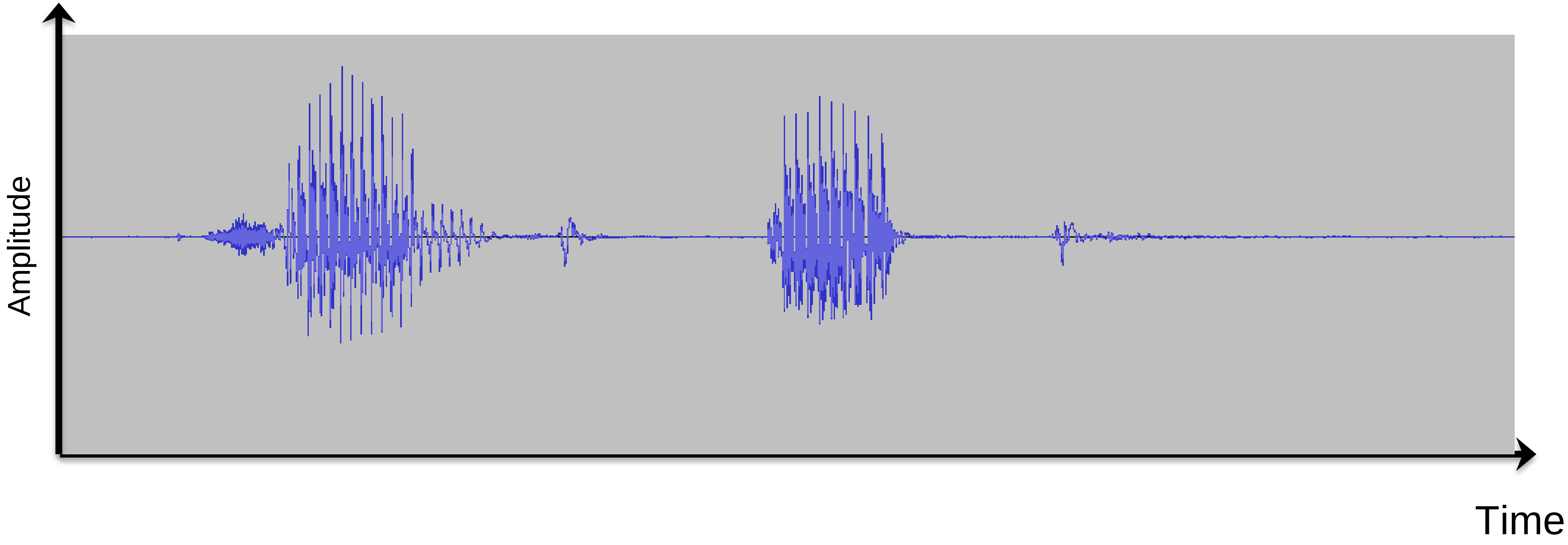
Speech recognition progress



- NIPS09/Hinton: **DBN** got 23% PER (from 27.4%) on TIMIT
- ICASSP2011 / Dahl, Dong Yu, Deng Li etc: 16% WERR over MPE HMM model on Bing VS task
- InterSpeech2011 / Frank Seide, Gang L, Dong Yu: **CD-DNN-HMM** got 32% WERR from 23.6% to 16.1% over discriminative trained HMM on SW
- ICASSP2013 / Alex Graves: **LSTM** 17.7% PER on TIMIT
- ICASSP2013 / Tara@IBM: Deep **CNN** 4-7% WERR on SW task
- InterSpeech2014 / Hasim Sak : SE **LSTMP** got 8% WERR on Google VS task (10.7% -> 9.8%)
- ICML-14 / Alex Graves: **E2E** learning with **CTC**
- ICASSP2015/Tara@IBM: **CLDNN** got 4-6% WERR on Google VS task over LSTM

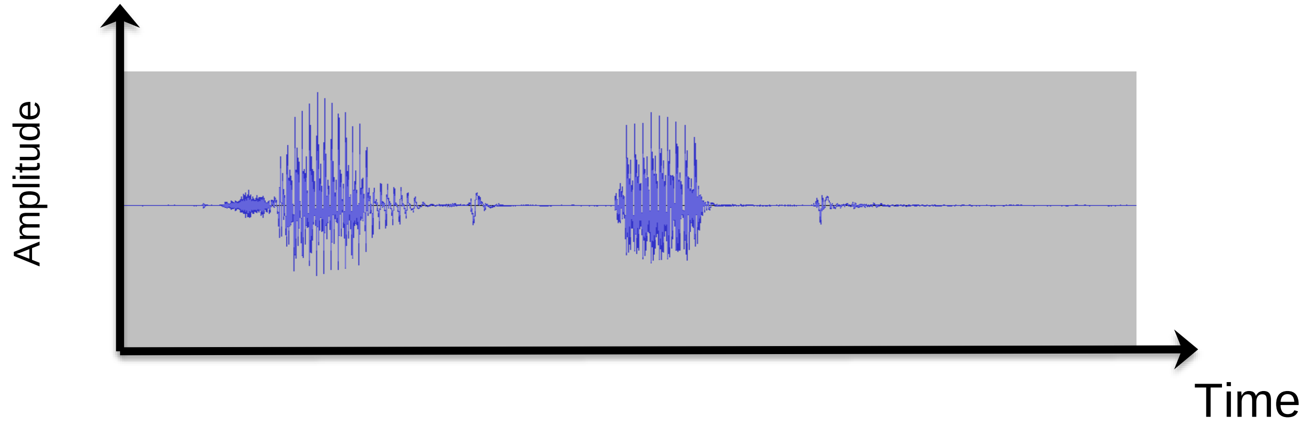
The Speech Recognition problem

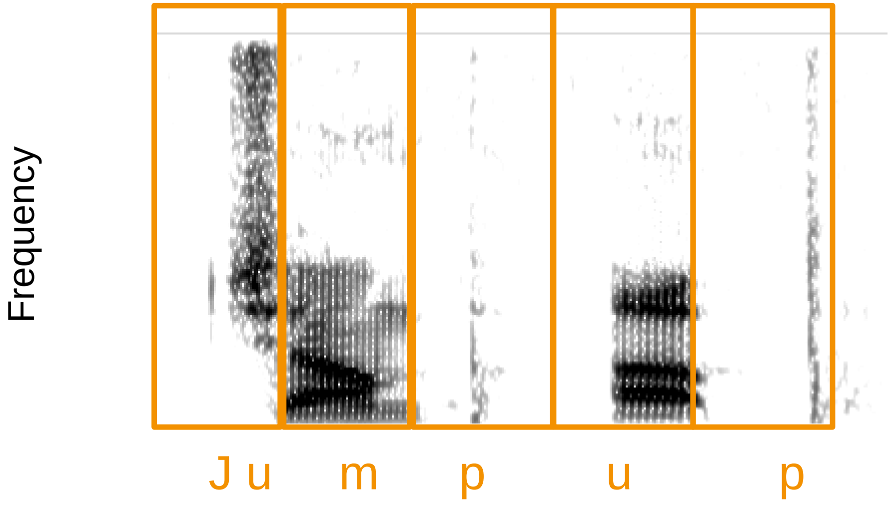
- Turn



- Into “______*Jump*_________ *up*__________________”
- And then into an action

A better way to look at it



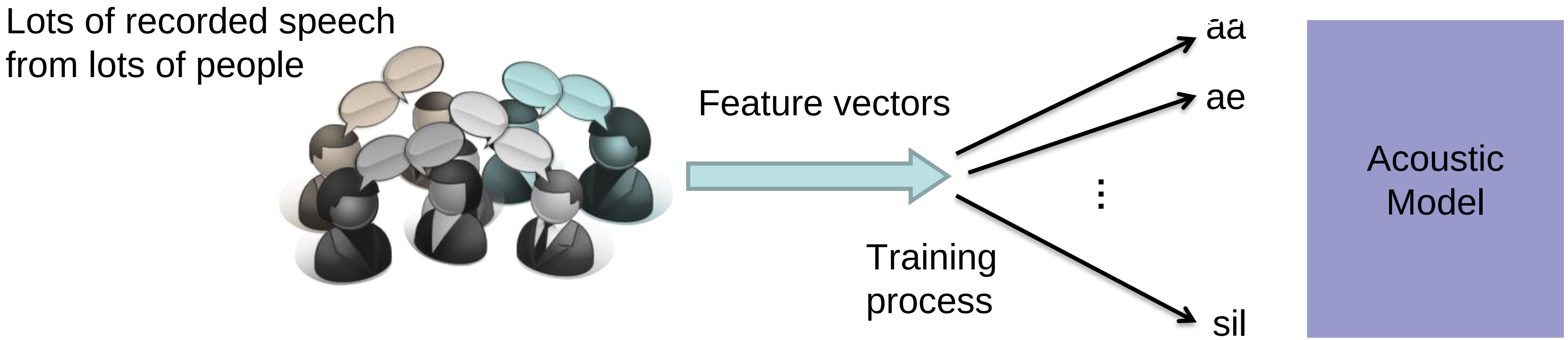


Front end

The front end extracts
Feature vectors
describing the salient features are derived from the signal every 10ms.

Acoustic Model

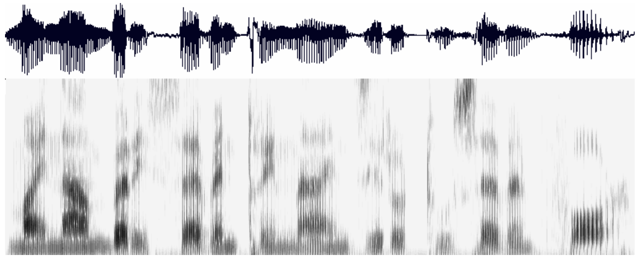
- Statistical representation of speech sounds of a language
- About 40 phonemes in US English
- Acoustic model -> likelihood that a feature vector was produced by a particular phoneme

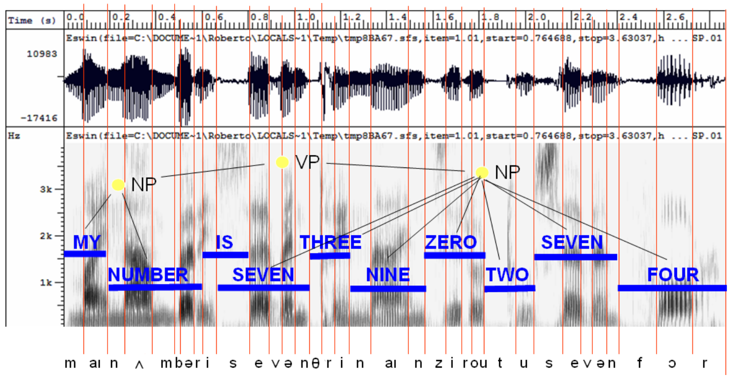


How might computers do it?

Acoustic waveform

Acoustic signal





Speech recognition

- Digitization
- Acoustic analysis of the speech signal
- Linguistic interpretation

An Example

Which sequence of speech sounds matches the best?

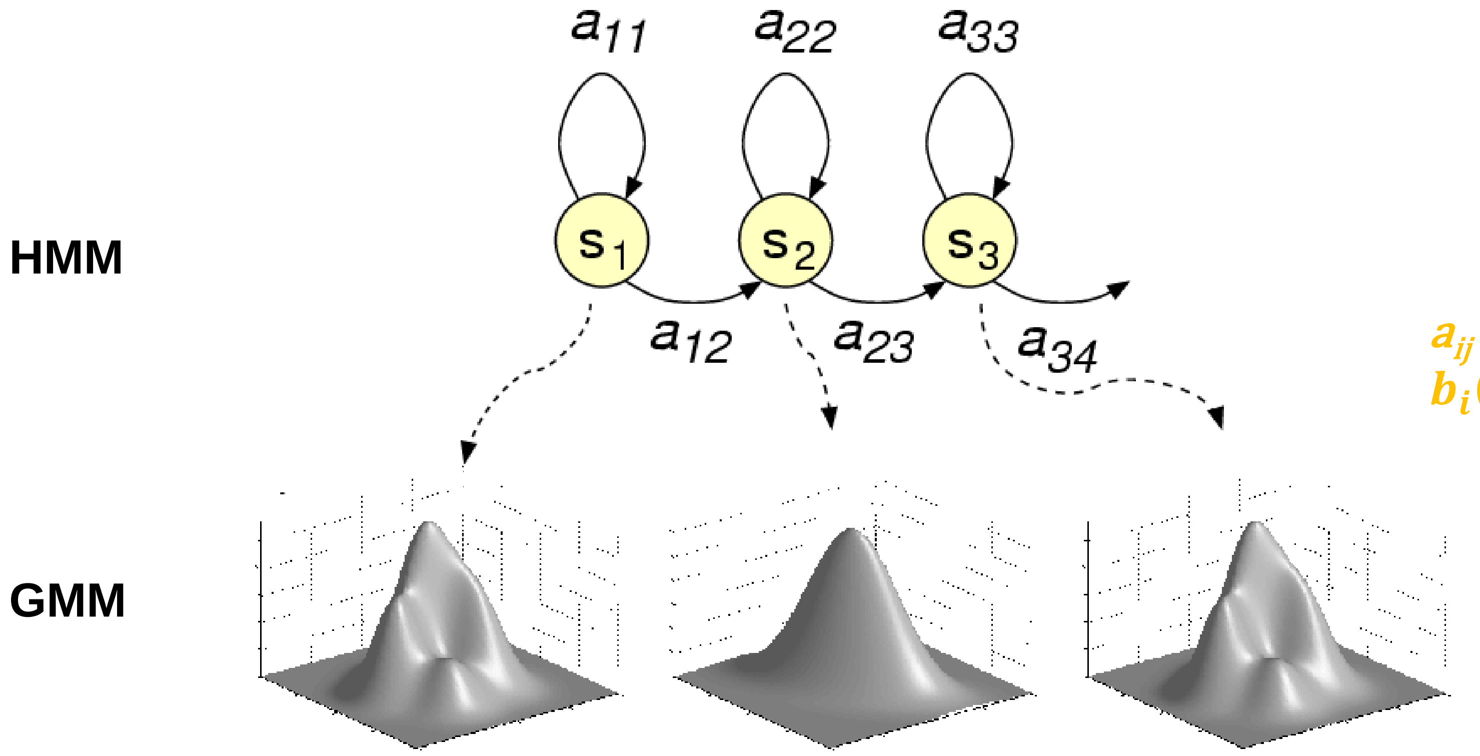
jump up –>sil JH AH M P AH P sil

dump cup –>sil D AH M P K AH P sil

open door-> sil O P AX N D O R sil

Legend: Good match, Fair match, Poor match

HMM



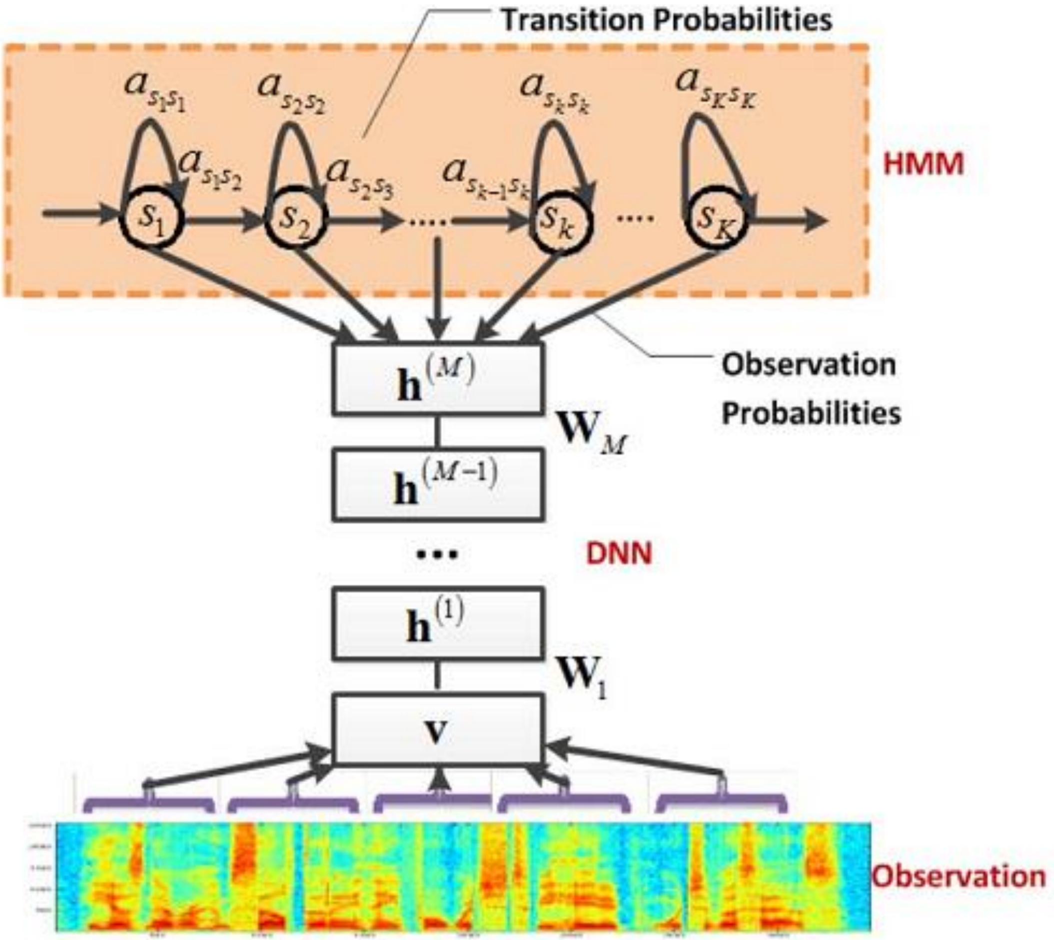
a_{ij}: **Transition probability**
$b_i(o)$: **Output probability**

HMM - to deal with the temporal variability of speech
GMM - to model the acoustic feature distribution in a short window (frame) for each HMM state

CD-DNN-HMM

- A hybrid system
- Combines the discriminative modeling power of DNN with the sequential modeling power of HMM.
- Uses DNN to compute the posterior probability of each HMM state
- Converting the p.p. into likelihood, which is used as acoustic score in decoder.

$$\boldsymbol{log}\left(\boldsymbol{P}(\boldsymbol{o}_t|\boldsymbol{s}_t)\right)$$
$$= \boldsymbol{log}\left(\boldsymbol{P}(\boldsymbol{s}_t|\boldsymbol{o}_t)\right) + \boldsymbol{log}\left(\boldsymbol{P}(\boldsymbol{o}_t)\right) - \boldsymbol{log}\left(\boldsymbol{P}(\boldsymbol{st})\right)$$



George Dahl, Dong Yu, Li Deng etc. 2011

Language Model

- The Language Model describes the probability of words and word sequences.

  So

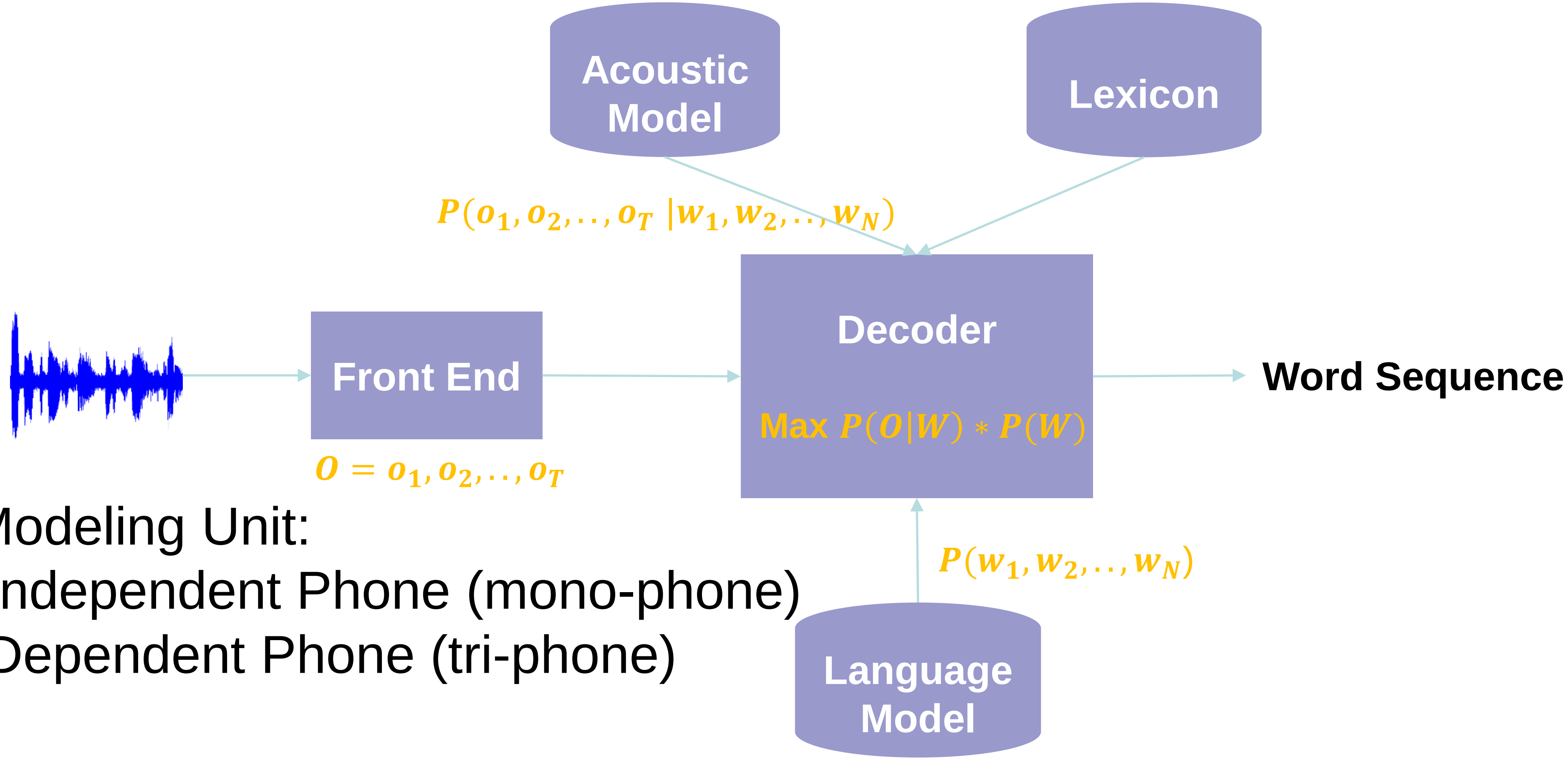
  "the" is much more probable than "kumquat"

  "drive a car" much more likely than "a car walk"

- Generate using lots of text data (500M+ sentences).

- Language Model probability can depend on context
  - Domain specific (vocabulary)

Speech Recognition System



Acoustic Modeling Unit:

- Context Independent Phone (mono-phone)
- Context Dependent Phone (tri-phone)

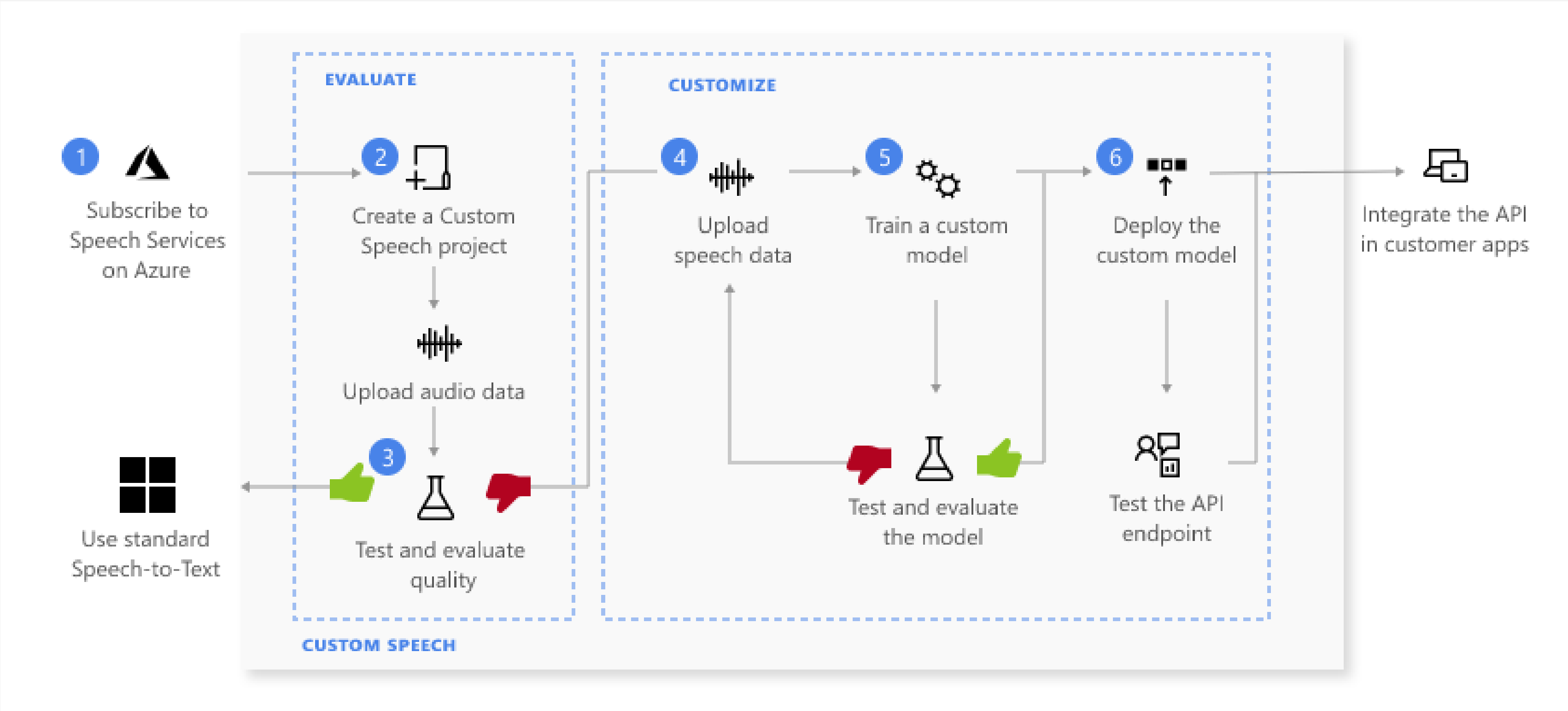
Speech Customization

- **Problem with Speech APIs**
  - Customers may have very specific needs
  - - Specific ages, accents, noise conditions -
  - General purpose endpoints can not be optimized

- **Solution**
  - Package speech recognition customization as a simple service
  - Data in, models out. Minimal configuration.
  - Language Model and Acoustic Model adaptation.
  - Allow third parties to create speech recognition endpoints.

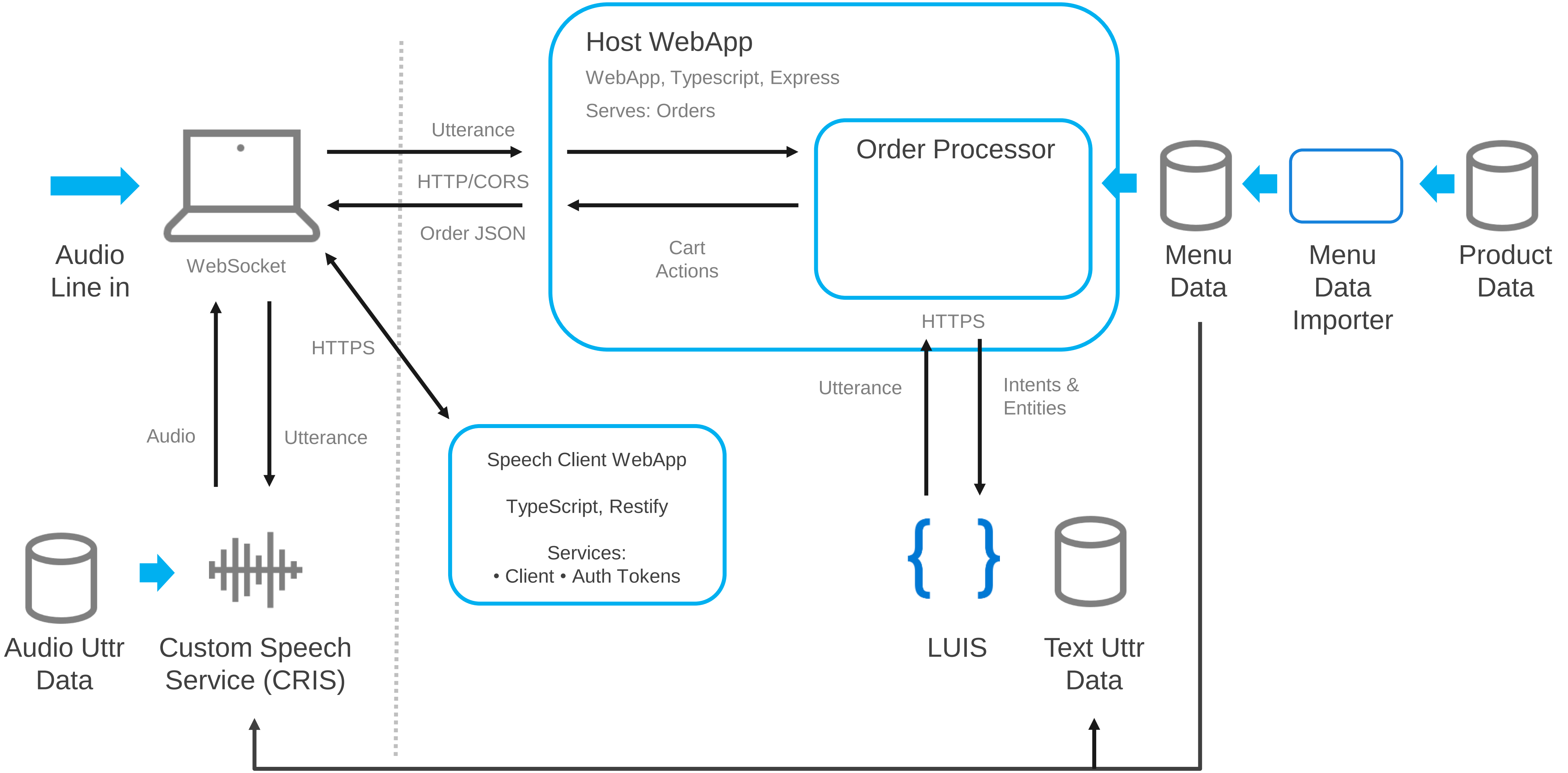
Speech Customization

Technische Hochschule Rosenheim



1. Create a Speech resource at portal.azure.com
2. Prepare some test audio files
3. Test Microsoft's standard speech-to-text
4. Upload training data (related text/audio/human transcripts)
5. Train custom speech-to-text model
6. Deploy model

Pizza Ordering Architecture



Applications

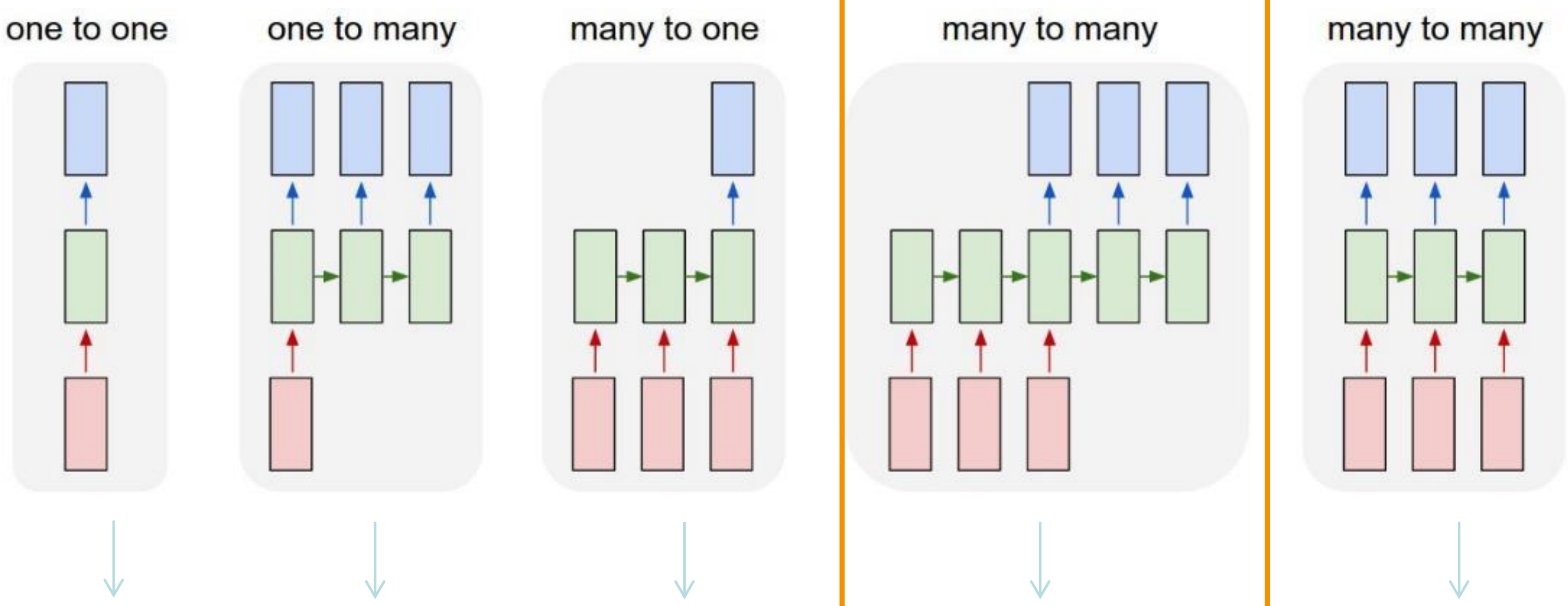


Image classification

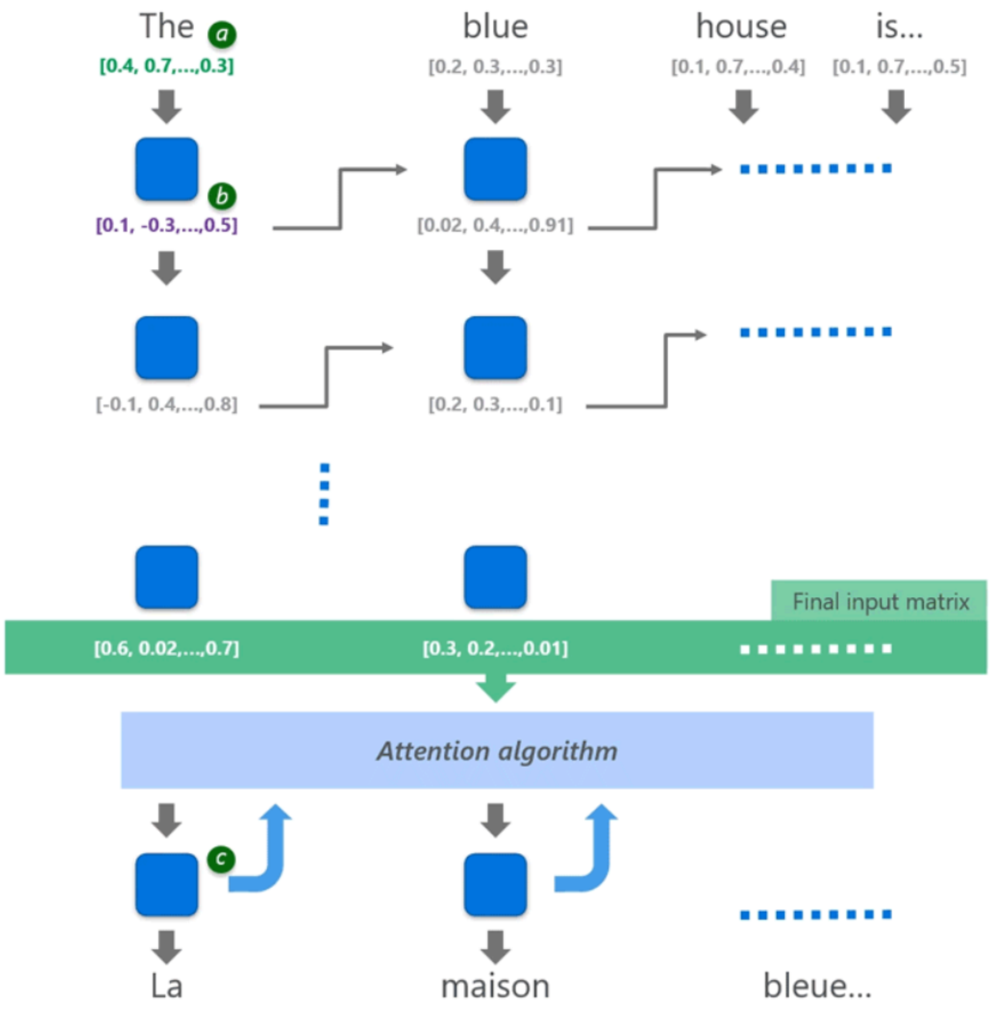
Image captioning

Sentiment analysis

Machine translation

Video classification

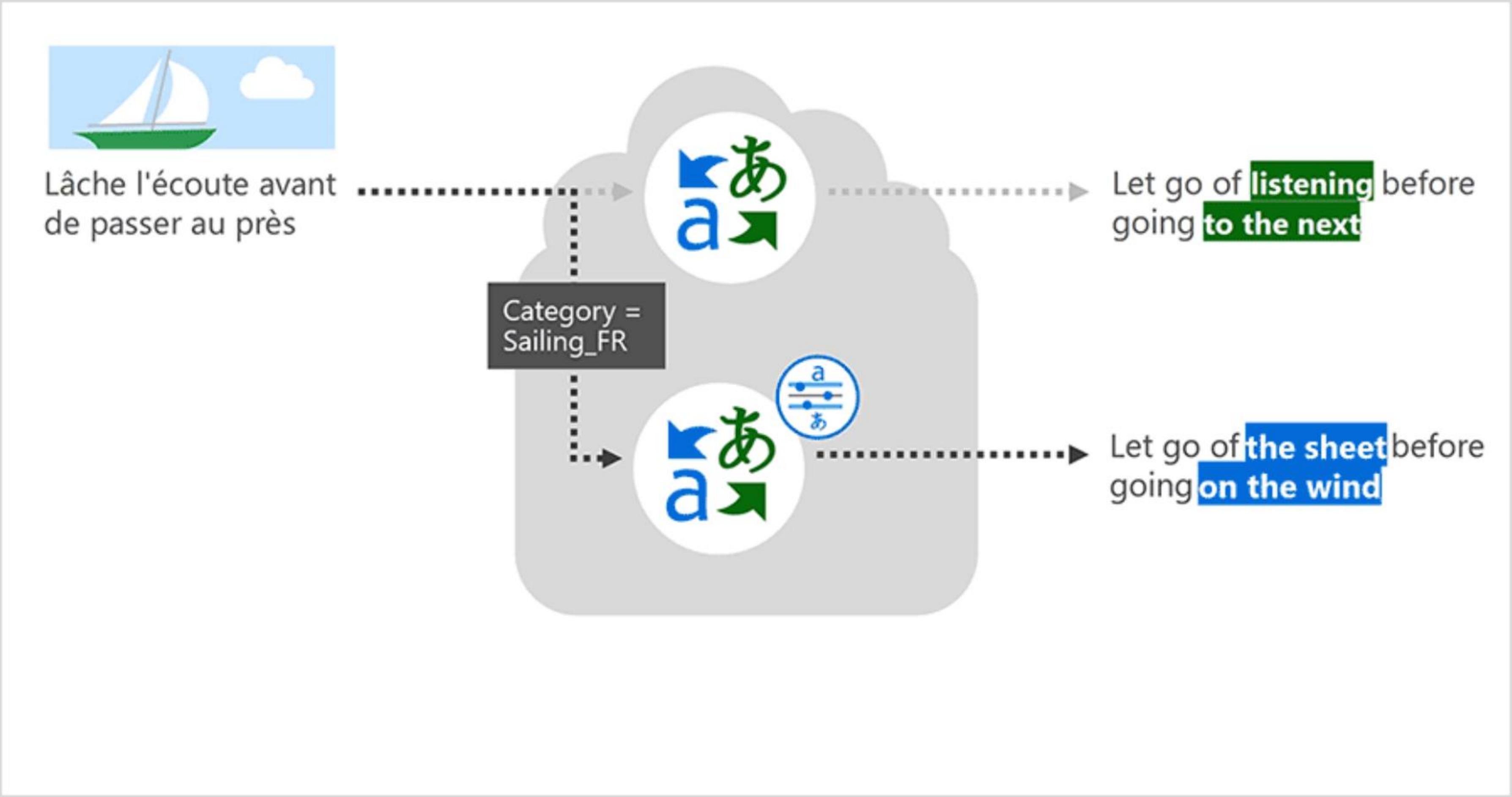
Text Translation



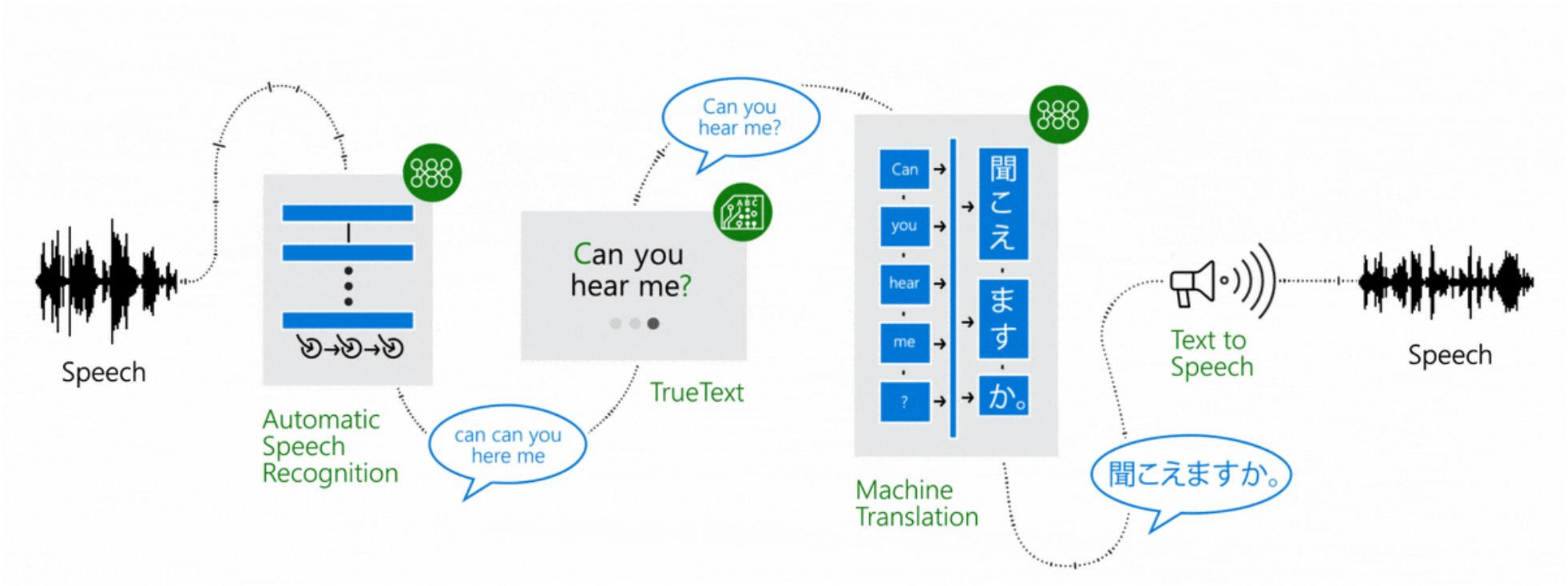
Use an RNN to sequentially predict the following word by taking the history into account!

WMT19 results: http://www.statmt.org/wmt19/pdf/53/WMT01.pdf

Customizing text translation to reflect domain-specific terminology



Speech Translation



References

- A Review of Keyphrase Extraction (https://arxiv.org/pdf/1905.05044.pdf)
- Findings of the 2019 Conference on Machine Translation (http://www.statmt.org/wmt19/pdf/53/WMT01.pdf)
- Key2Vec: Automatic Ranked Keyphrase Extraction from Scientific Articles using Phrase Embeddings