Chatbots models, NLU & ASR

Pierre Lison

IN4080: Natural Language Processing (Fall 2022)

18.10.2022

Plan for today

- Obligatory assignment
- Chatbot models (cont'd)
- Natural Language Understanding (NLU) for dialogue systems
- Speech recognition

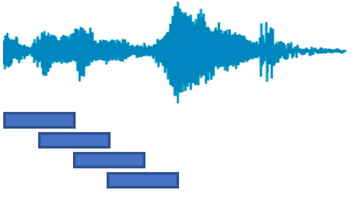
Plan for today

- **Obligatory assignment**
- Chatbot models (cont'd)
- Natural Language Understanding (NLU) for dialogue systems
- Speech recognition

Oblig 3

Three parts:

1. Chatbot trained on movie and TV subtitles
2. Silence detector in audio files
3. (Simulated) talking elevator



Oblig 3

- Deadline: November 11
  - Concrete delivery: **Jupyter notebook**
  - Text explanations in the notebook as important as the code itself!
- Don’t hesitate to ask questions during the group sessions
  - we are here to help!

Plan for today

- Obligatory assignment
- **Chatbot models (cont'd)**
- Natural Language Understanding (NLU) for dialogue systems
- Speech recognition

Chatbot models: recap

- Rule-based models:

```
if (some pattern match X on user input)
then respond Y to user
```

- IR models using cosine similarities between vectors

$$r = response\left(\underset{t \in C}{\operatorname{argmax}} \frac{q^T t}{||q||t||}\right)$$

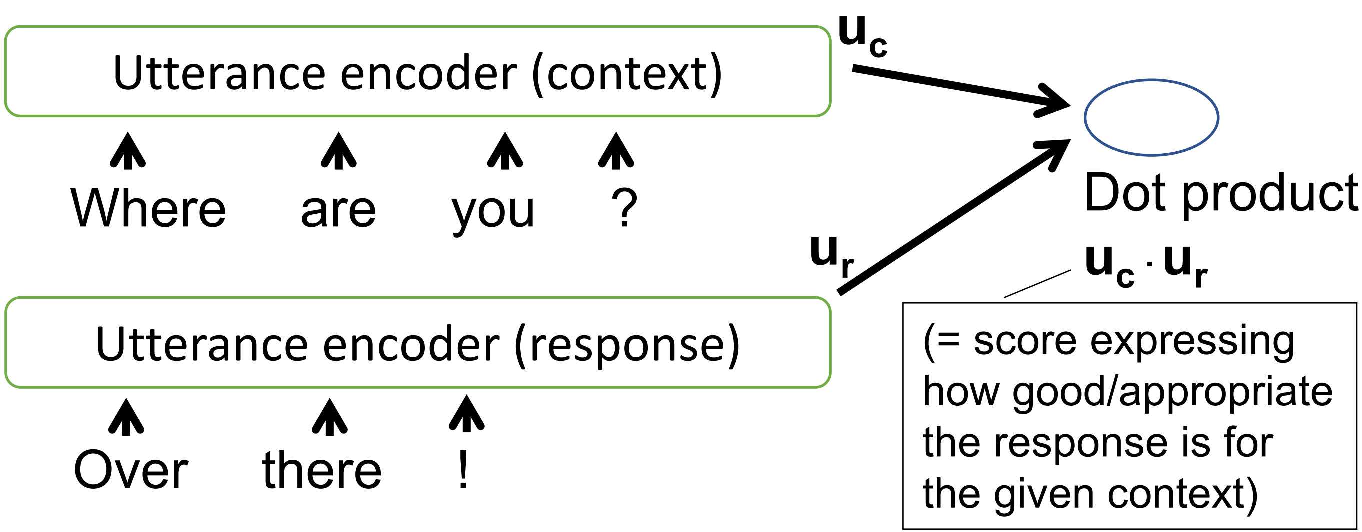
Where C is the set of utterances in dialogue corpus (in a vector representation)

and q is the user input (also in vector form)

Dual encoders

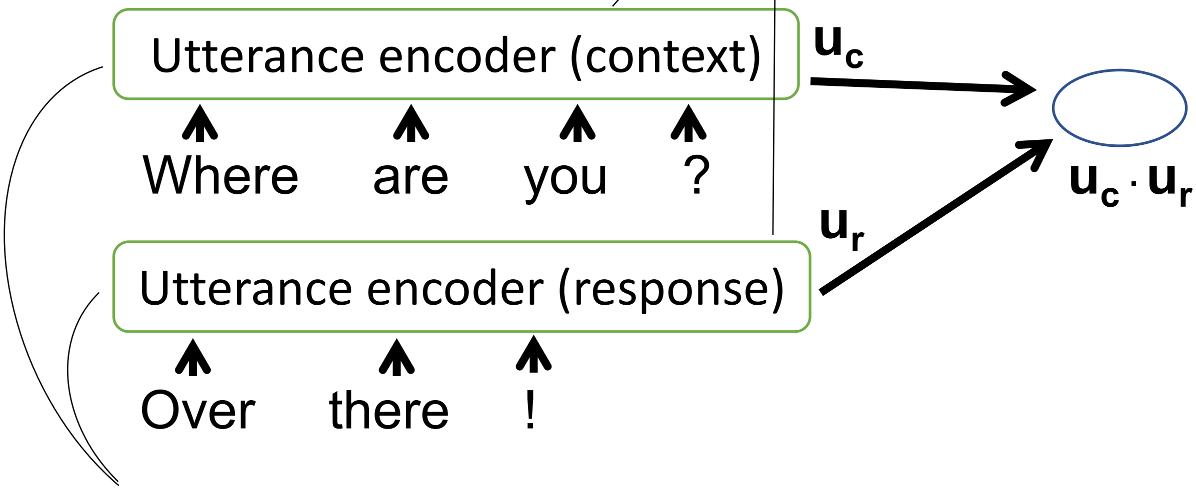
Another type of IR-based chatbots

- We compute here the dot product between the user input (called "*context*") and a possible *response*



Dual encoders

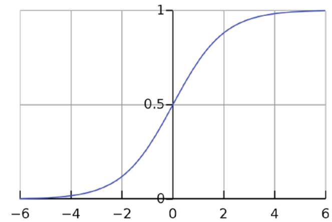
The encoders are typically deep neural networks based on e.g. transformers

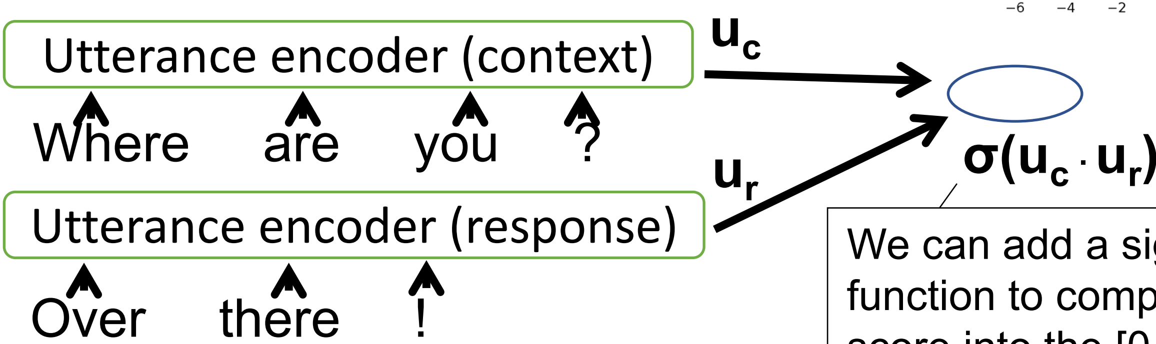


The two encoders often rely on a shared neural network, apart from a last transformation step that is specific for the context or response

Dual encoders

$$\sigma(x) = \frac{1}{1+e^{-x}}$$





We can add a sigmoid function to compress the score into the [0,1] range

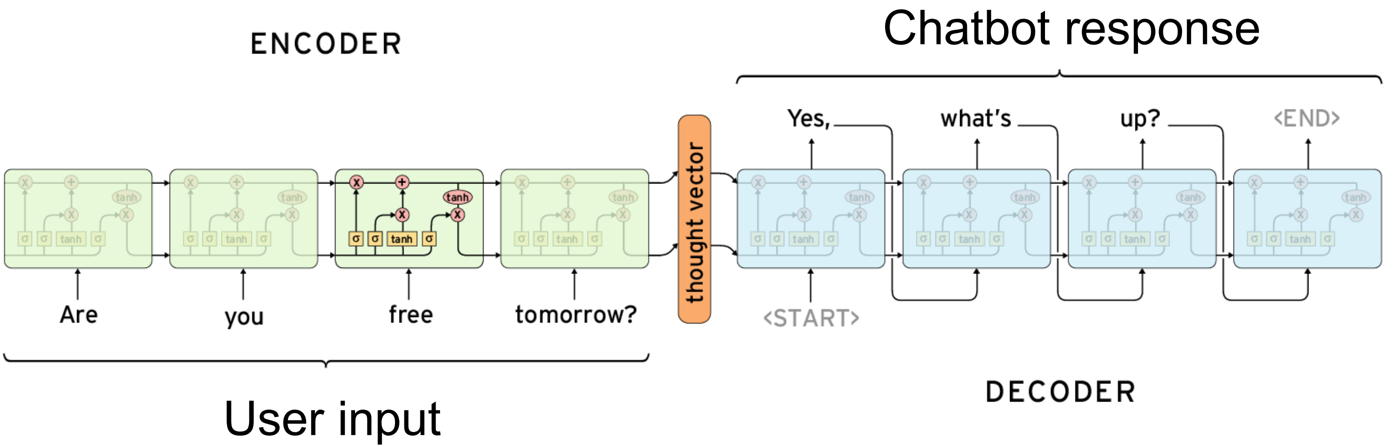
Dual encoders are trained with both *positive* and *negative* examples:

- *Positive* : actual consecutive pairs of utterances observed in the corpus → output=1
- *Negative*: random pairs of utterances → output=0

Seq2seq models

- Sequence-to-sequence models *generate* a response token-by-token
  - Akin to machine translation
  - Can generate new responses never observed in the corpus
- Two steps:
  - First «encode» the input with a neural model (=tokenise the input and extract the vectors for each token)
  - Then «decode» the output token-by-token (based on the input vectors and the output produced so far)

Seq2seq models



NB: state-of-the-art seq2seq models use an attention mechanism (not shown here) above the recurrent layer

[Image borrowed from Deep Learning for Chatbots: Part 1]

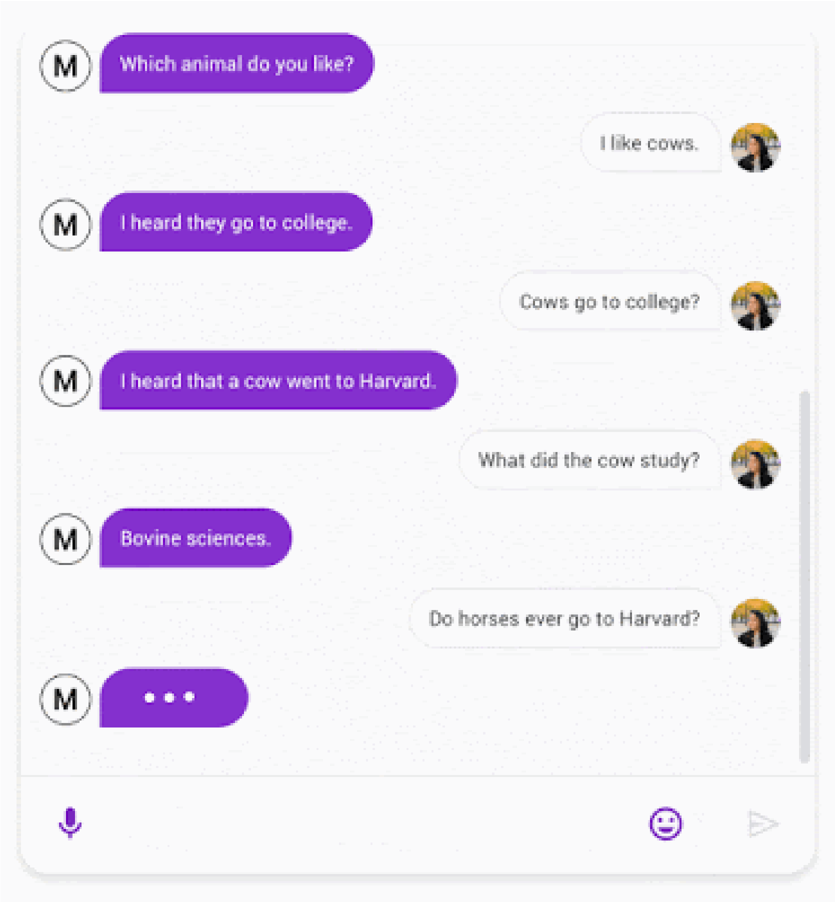
Seq2seq models

- Interesting models for dialogue research
- **But**:
  - Difficult to «control» (hard to know in advance what the system may generate)
  - Lack of diversity in the responses (often stick to generic answers: «I don’t know» etc.)
  - Getting a seq2seq model that works reasonably well takes time (and often lots of data)

[Li, Jiwei, et al. (2015) "A diversity-promoting objective function for neural conversation models.», ACL]

Example from Meena (Google)

2.6 billion parameters, trained on 341 GB of text (public domain social media conversations)



https://ai.googleblog.com/2020/01/towards-conversational-agent-that-can.html

Taking stock

- Rule-based chatbots
- Corpus-based chatbots
  - IR approaches
  - seq2seq

Pro: Fine-grained control on interaction

Con: Difficult to build, scale and maintain

Pro: Easy to build, well-formed responses

Con: Can only repeat existing responses in corpus

Pro: Powerful model, can generate anything

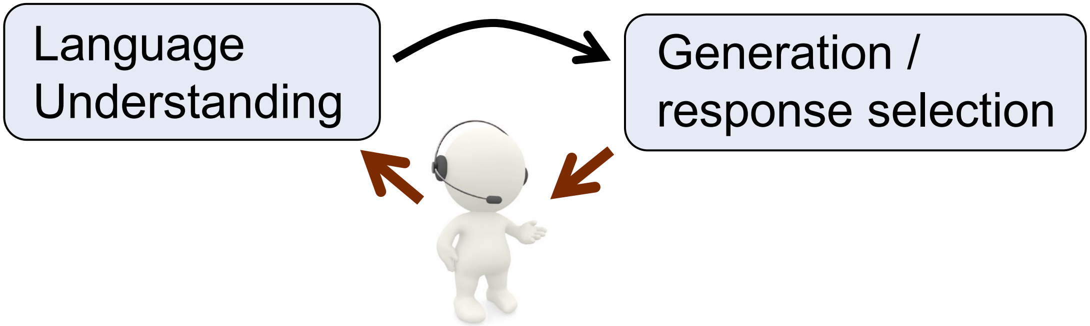
Con: Difficult to train, hard to control, needs lots of data

Corpus-based approaches seen so far often limited to chi-chat dialogues (for which we can easily crawl data)

Plan for today

- Obligatory assignment
- Chatbot models (cont'd)
- **Natural Language Understanding (NLU) for dialogue systems**
- Speech recognition

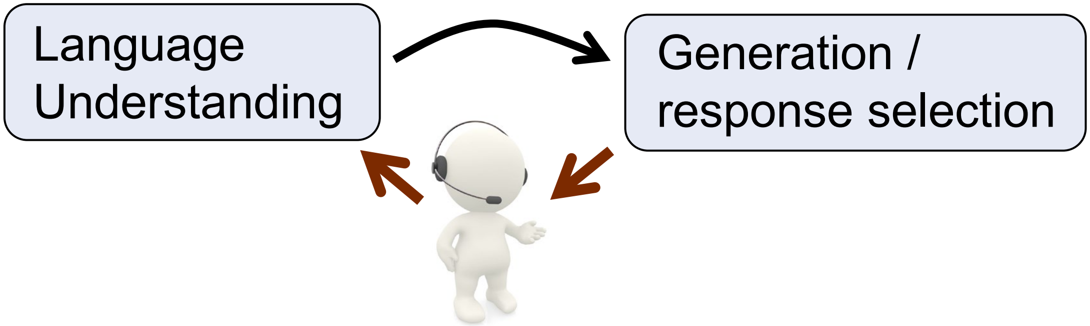
NLU-based chatbots



Can we build data-driven chatbots for task-specific interactions (not just chit-chat)?

- "Standard" case for commercial chatbots
- Typically: no available task-specific dialogue data

NLU-based chatbots

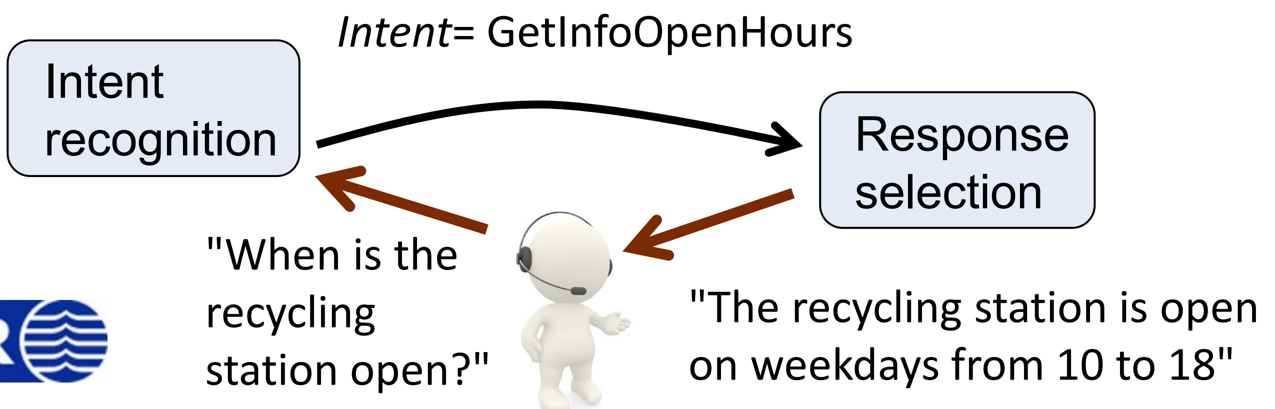


- Solution: NLU as a **classification task**
  - From a set of (predefined) possible **intents**
- Response selection generally handcrafted
  - Chatbot owners want to have control over what the chatbot actually says

Intent recognition

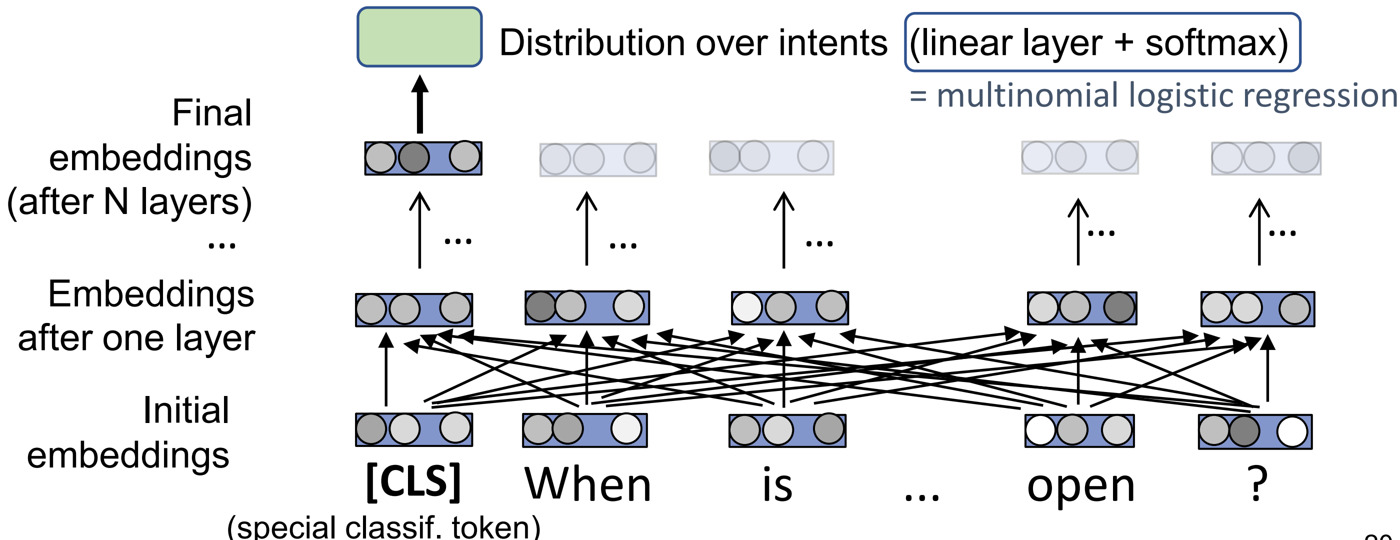
Goal: map user utterance to its most likely intent

- *Input*: sequence (of characters or tokens)
  + possibly preceding context
- *Output*: intent (what the user tries to accomplish)



Intent recognition

- Many possible machine learning models
  - Convolutional, recurrent, transformers, etc
- Example using BERT:



Intent recognition

- Need to collect *training data* to learn this classification model
  - *Data*: user utterances (+ context) manually annotated with their intent(s)
  - Often annotated by "chatbot trainers" in industry
- Standard approach these days:
  - Take a pre-trained neural language model (i.e. NorBERT for Norwegian)
  - *Fine-tune* it for this specific classification task

Slot filling

- In addition to intents, we also sometimes need to detect specific entities ("slots"), such as mentions of places or times

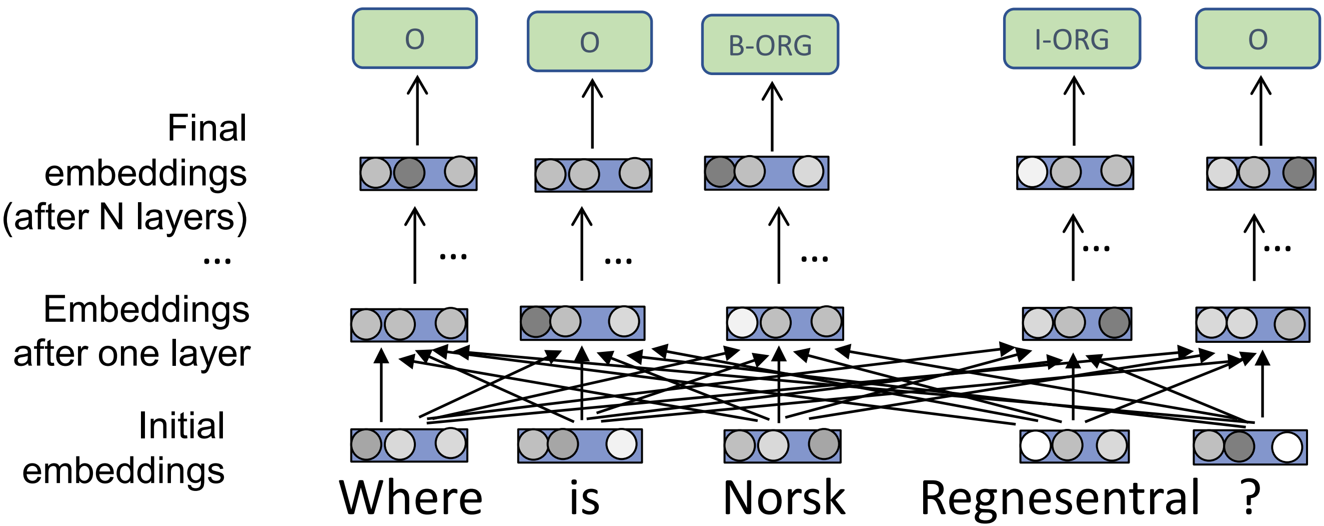
«Show me morning flights from Boston to San Francisco on Tuesday» →

DOMAIN:	AIR-TRAVEL
INTENT:	SHOW-FLIGHTS
ORIGIN-CITY:	Boston
ORIGIN-DATE:	Tuesday
ORIGIN-TIME:	morning
DEST-CITY:	San Francisco

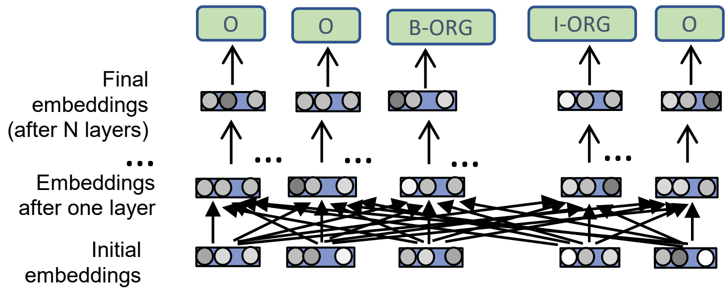
- Slots are domain-specific
  - And so are the ontologies listing all possible values for each slot

Slot filling

Can be framed as a *sequence labelling task* (as in NER), using e.g. **BIO** schemes



Slot filling



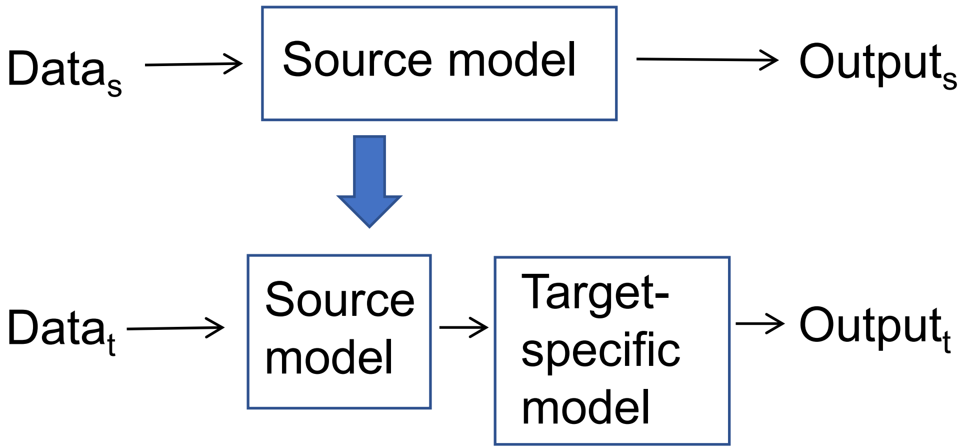
- Token-level classification task
  - Output classes: BIO-prefixed categories
- Slot-filling models also need to be trained / fine-tuned on annotated training data
- Possible to fine-tune intent classifier and slot filler on same model

Small amounts of data?

1. Use *transfer learning* to exploit models trained on related domains

Source domain (with large amounts of training data)

Target domain (with small amounts of training data)



Fine-tuning of a pre-existing language model is a type of transfer learning

Small amounts of data?

1. Use *transfer learning* to exploit models trained on related domains

2. Use *data augmentation* to generate new labelled utterances from existing ones

"**When** is the recycling station open?" → GetInfoOpenHours

⇩ Replace with synonyms

"**At what time** is the recycling station open?" → GetInfoOpenHours

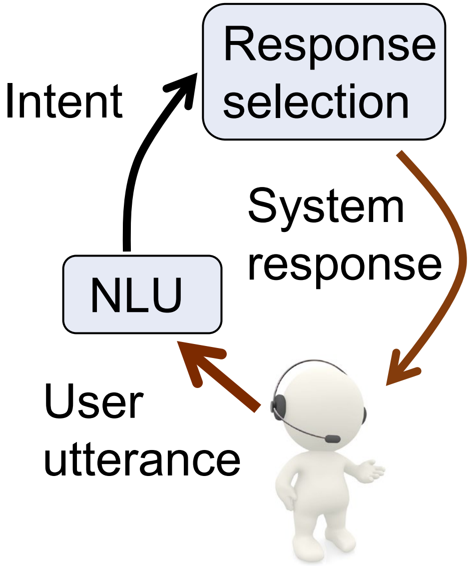
Small amounts of data?

1. Use *transfer learning* to exploit models trained on related domains
2. Use *data augmentation* to generate more utterances from existing ones
3. *Label more data*, either manually or using weak supervision techniques

[see e.g. Mallinar et al (2019), "Bootstrapping conversational agents with weak supervision", IAAI.]

Response selection

- Given an intent, how to create a response?
- In commercial systems, system responses are typically written by hand
  - Possibly in templated form, i.e. "{Place} is open from {Start-time} to {Close-time}"
- But data-driven generation methods also exists

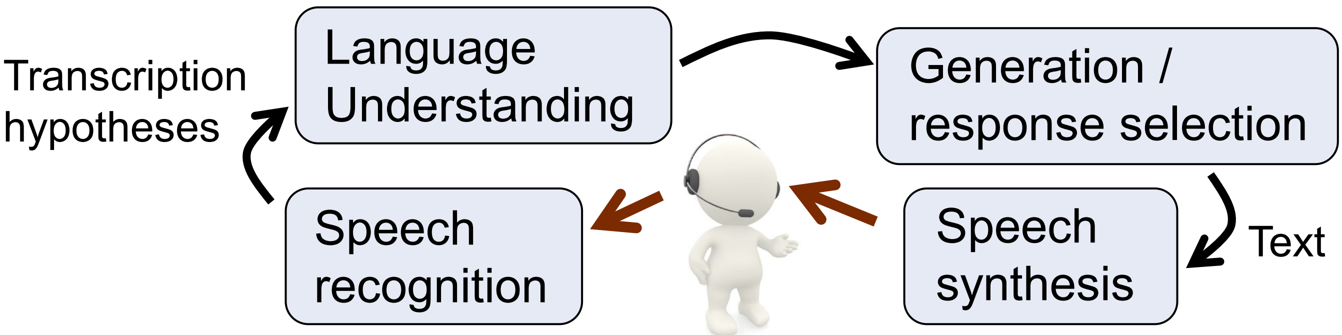


[see e.g. Garbacea & Mei (2020), "*Neural Language Generation: Formulation, Methods, and Evaluation*"]

Plan for today

- Obligatory assignment
- Chatbot models (cont'd)
- Natural Language Understanding (NLU) for dialogue systems
- **Speech recognition**

Spoken dialogue systems

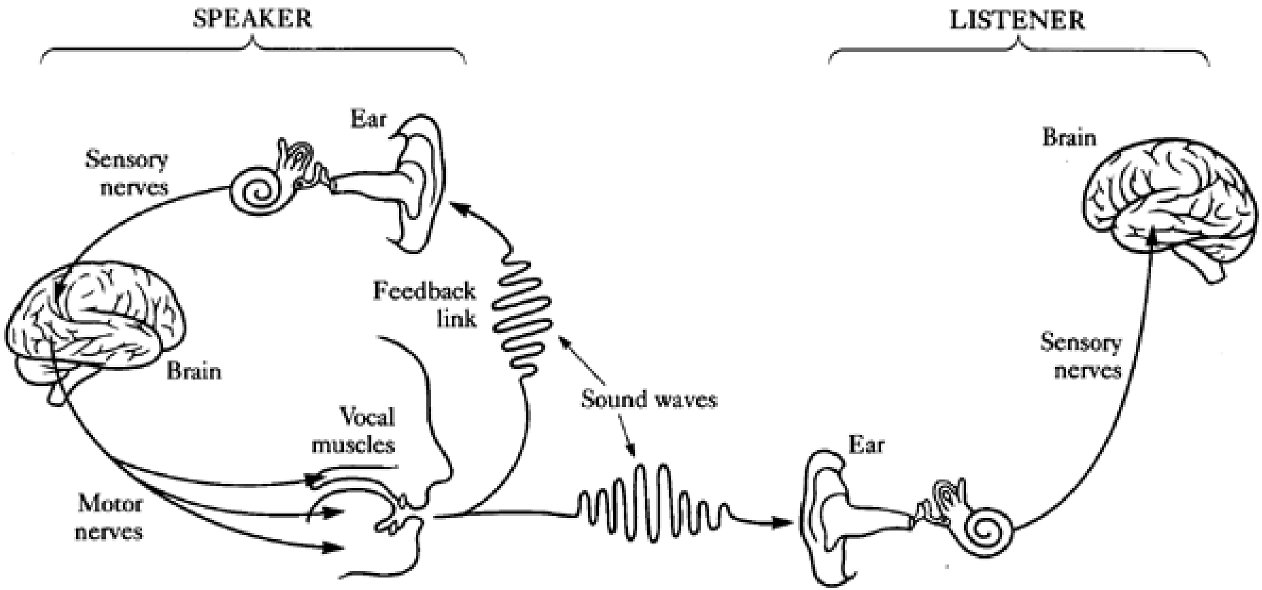


Spoken interfaces add a layer of complexity

- Need to handle uncertainties, ASR errors etc.
- Speech communicates more than just words (intonation, emotions in voice, etc.)
- Need to handle turn-taking

A difficult problem!

The speech chain



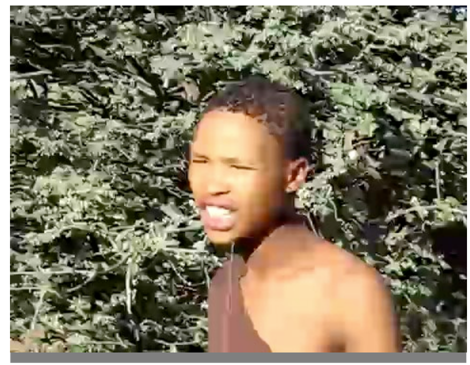
[Denes and Pinson (1993), «The speech chain»]

Speech production

- ► Sounds are *variations in air pressure*
- ► How are they produced?
  - An **air supply**: the *lungs* (we usually speak by breathing out)
  - A **sound source** setting the air in motion (e.g. vibrating) in ways relevant to speech production: the *larynx*, in which the *vocal folds* are located
  - A set of 3 **filters** modulating the sound: the *pharynx*, the *oral tract* (teeth, tongue, palate,lips, etc.) & the *nasal tract*

Speech production

Visualisation of the vocal tract via *magnetic resonance imaging* [MRI]:

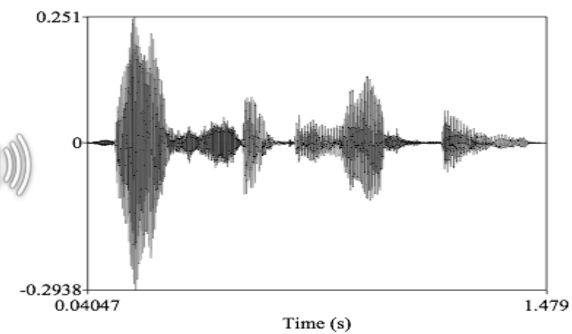


NB: A few languages also rely on sounds not produced by vibration of vocal folds, such as *click languages* (e.g. Khoisan family in south-east Africa):

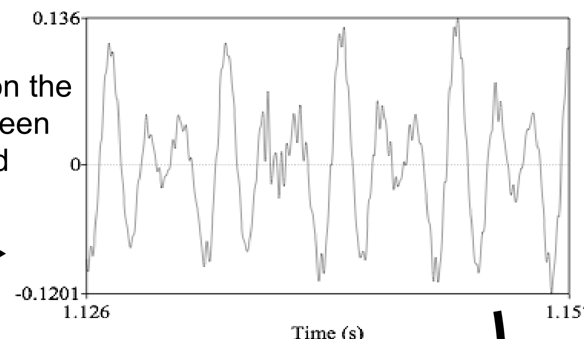
Speech perception

A (speech) sound is *a variation of air pressure*

- This variation originates from the speaker's speech organs
- We can plot a *wave* showing the changes in air pressure over time (zero value being the normal air pressure)



zoom on the part between 1.126 and 1.157 s.



About 4 cycles in the waveform, which means a frequency of about 4/0.03 ≈129 Hz

Important measures

1. The **fundamental frequency F_0**: lowest frequency of the sound wave, corresponding to the speed of vibration of the vocal folds (between 85-180 Hz for male voices and 165-255 Hz for female voices)

2. The **intensity**: the signal power normalised to the human auditory threshold, measured in **dB** (decibels):

$$\text{Intensity} = 10\ \log_{10} \frac{\text{Power}}{P_0} = 10\ \log_{10} \frac{1}{N P_0} \boxed{\sum_{i=1}^{N} y(t_i)^2}$$

Total energy of signal

for a sample of N time points $t_1, \ldots t_N$
P_0 is the human auditory threshold, = 2×10^{-5} Pa

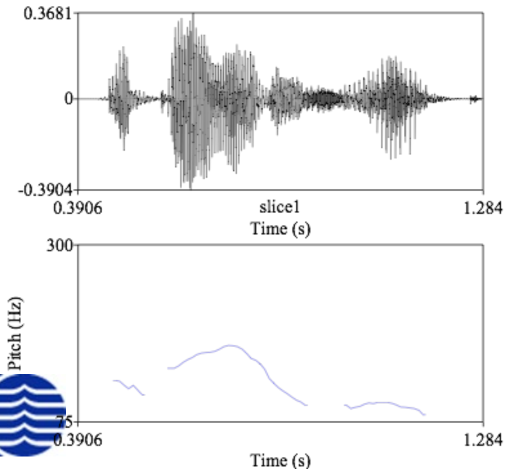
Note: dB scale is logarithmic, not linear!

Why are F_0 and the intensity important?

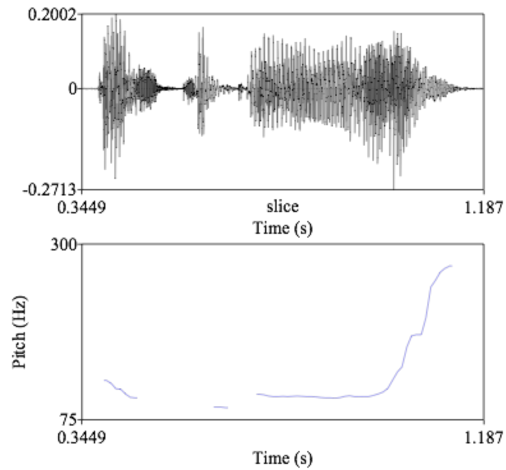
F_0 correlates with the *pitch* of the voice, and the pitch movement for an utterance will give us its *intonation*

Interrogative utterance
= rising intonation at the end

"The ball is red"



"Is the ball red?"

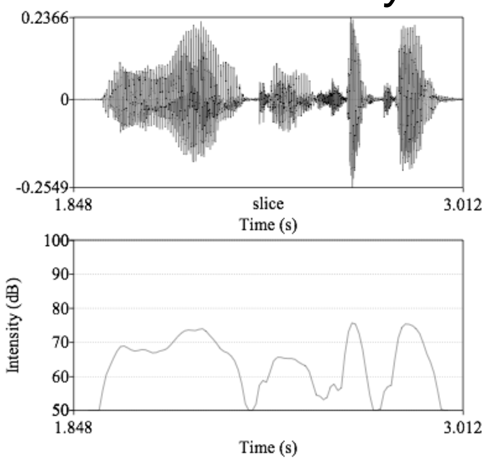


Why are F_0 and the intensity important?

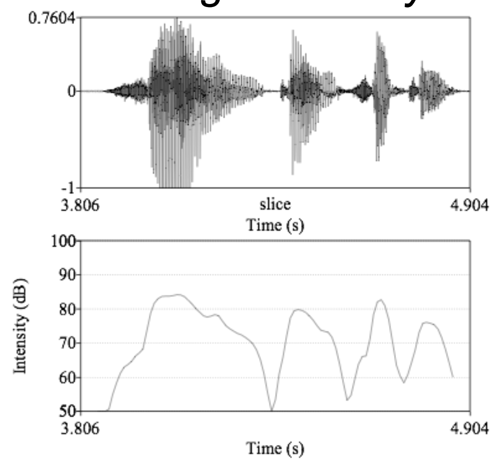
F_0 correlates with the *pitch* of the voice, and the pitch movement for an utterance will give us its *intonation*

The signal intensity corresponds to the *loudness* of the speech sound

low intensity

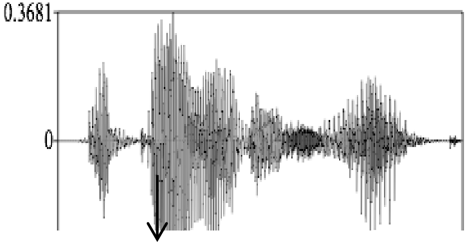


high intensity



The speech recognition task

Input: Audio data



Sequence **O** of acoustic observations (i.e. every 20 ms)

Output: Transcription

"The ball is red"

Goal: Map speech signal **O** into sequence of linguistic symbols $\widehat{W}$ (words or characters):

$$\hat{W} = \underset{W}{\operatorname{argmax}}\, P(W|O)$$

Why is ASR difficult?

- *Many sources of variation*: speaker voice (and style), accents, ambient noise, etc.

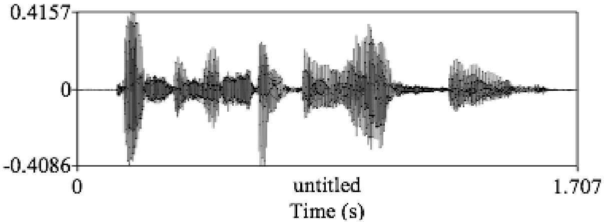
Why is ASR difficult?

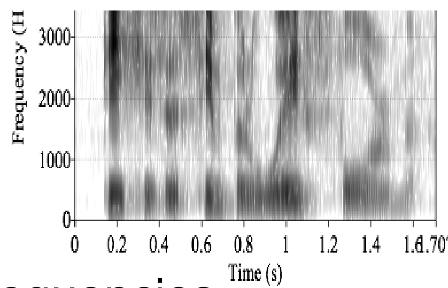
- *Many sources of variation*: speaker voice and speaking style, accents, ambient noise, etc.
- Very long input sequences
  - For audio frames lasting 20 ms. and offset of 10 ms. → 100 observations per sec. (each observation including many numerical features)

- But output sequence (e.g. phonemes, characters or tokens) much shorter and ***no explicit alignment between input and output***

Preprocessing

- Most speech sounds cannot be distinguished from the raw waveform
- Better: convert the signal to a representation of the signal's *component frequencies*
  - Based on Fourier's transform





spectrogram showing which frequencies are most active at a given time

"Classical" model

Using Bayes' rule, we can rewrite Ŵ as:

$$\hat{W} = \underset{W}{\operatorname{argmax}} \frac{P(O|W)P(W)}{P(O)}$$ (Bayes)

$$= \underset{W}{\operatorname{argmax}} P(O|W)P(W)$$ (P(O) constant for all W)

Acoustic model

Language model

Determines the probability of the acoustic inputs O given the word sequence W

Determines the probability of the word sequence W

Neural ASR

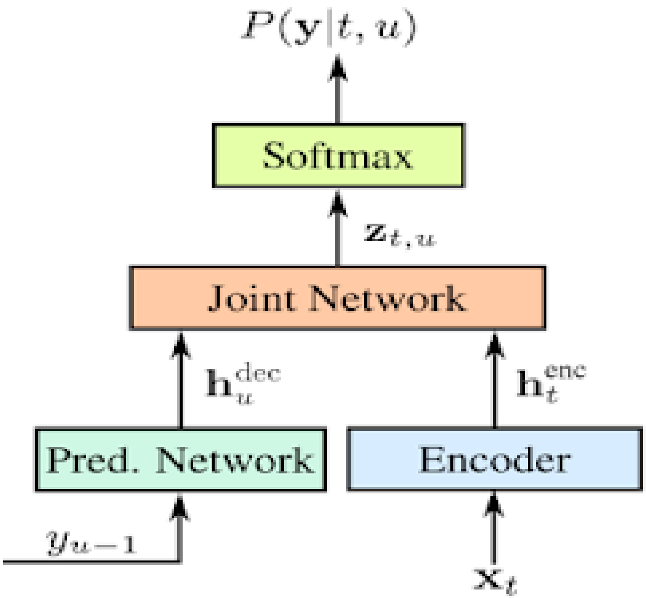
- The best performing ASR are *deep, end-to-end neural architectures*
  - Less dependent on external ressources (such as pronunciation dictionaries)
  - Move from carefully handcrafted acoustic features to *learned* representations
- Too time demanding to review here
  - But they rely on the same building blocks as other NNs: convolutions, recurrence, (self-)attention, etc.

Neural ASR

https://ai.googleblog.com/2019/03/an-all-neural-on-device-speech.html

An example of a relatively simple neural model: Google's on-device ASR

- *Encoder* maps audio signal $\mathbf{x}_t$ to hidden representations (with stacked LSTMs)
- *Prediction Network* is a language model
- Model then merges the two hidden representations and predicts outputs character-by-character



ASR Performance

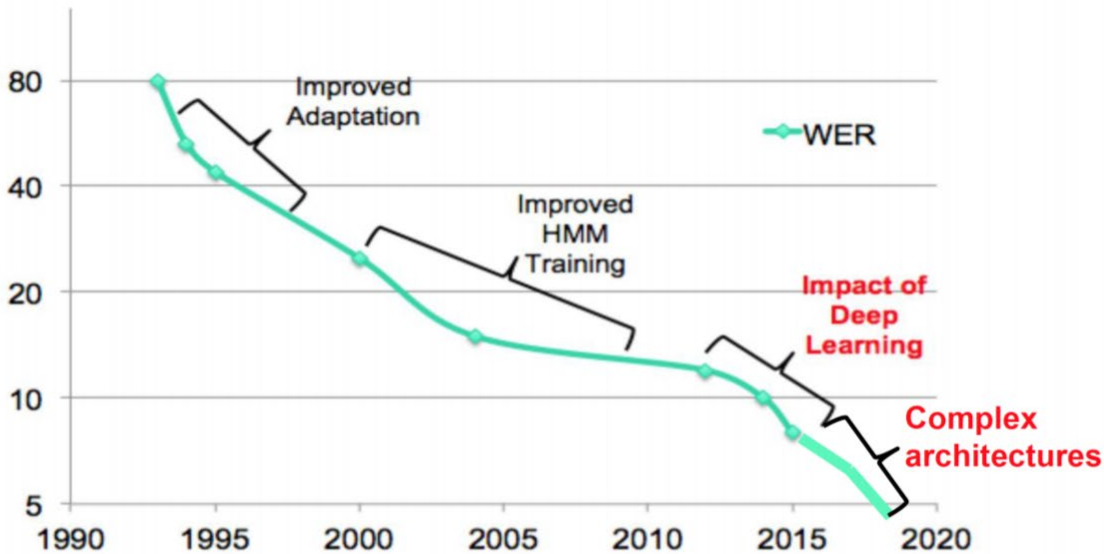


Figure: ASR Performance[1] on English Conversational Telephony (Switchboard)

[Figure from Bhuvana Ramabhadran's presentation at Interspeech 2018]

ASR evaluation

- Standard metric: **Word Error Rate**
  - Measures how much the utterance hypothesis h differs from the «gold standard» transcription t*
- = Minimum edit distance between h and t*, counting the number of word substitutions, insertions and deletions:

$$\text{Word Error Rate} = 100 \times \frac{\text{Insertions} + \text{Substitutions} + \text{Deletions}}{\text{Number of words in transcription}}$$

ASR evaluation

Gold standard Transcription	*yes can you now rotate this triangle*
ASR hypothesis	*yes can you not rotate this triangle there*

$$\mathrm{WER} = 100 \times \frac{1\ \text{Sub} + 1\ \text{Ins}}{7} = 28.6\%$$

Gold standard Transcription	*there is five and*
ASR hypothesis	*the size and*

$$\mathrm{WER} = 100 \times \frac{2\ \text{Sub} + 1\ \text{Del}}{4} = 75\%$$

Disfluencies

- Speakers construct their utterances «as they go», incrementally
  - Production leaves a *trace* in the speech stream
- Presence of multiple disfluencies
  - Pauses, fillers («øh», «um», «liksom»)
  - Repetitions («the the ball»)
  - Corrections («the ball err mug»)
  - Repairs («the bu/ ball»)

Disfluencies

Internal structure of a disfluency:

$$\text{Book a ticket } \underbrace{\text{to Boston}}_{\text{reparandum}} \; \underbrace{\text{uh I mean}}_{\text{interregnum}} \; \underbrace{\text{to Denver}}_{\text{repair}}$$

- reparandum: part of the utterance which is edited out
- interregnum: (optional) filler
- repair: part meant to replace the reparandum

[Shriberg (1994), «Preliminaries to a Theory of Speech Disfluencies», Ph.D thesis]

Some disfluencies

så gikk jeg e flytta vi til Nesøya da begynte jeg på barneskolen der
og så har jeg gått på Landøya ungdomsskole # som ligger ## rett over broa nesten # rett med Holmen

jeg gikk på Bryn e skole som lå rett ved der vi bodde den gangen e barneskole
videre på Hauger ungdomsskole

da hadde alle hele på skolen skulle liksom # spise julegrøt og det va- det var bare en mandel
og da var jeg som fikk den da ble skikkelig sånn " wow # jeg har fått den " ble så glad

[«*Norske talespråkskorpus - Oslo delen*» (NoTa), collected and annotated by the Tekstlaboratoriet]

Plan for today

- Obligatory assignment
- Chatbot models (cont'd)
- Natural Language Understanding (NLU)
- Speech recognition
- **Summary**

Summary

How to develop a chatbot:

- **Rule-based approaches**

Language Understanding

matched condition

Generation / response selection

= recognition of handcrafted patterns (e.g. regular expressions)

= handcrafted responses or templates to fill

Summary

How to develop a chatbot:

- Rule-based approaches
- **IR-based approaches**

Language Understanding

embedding

Generation / response selection

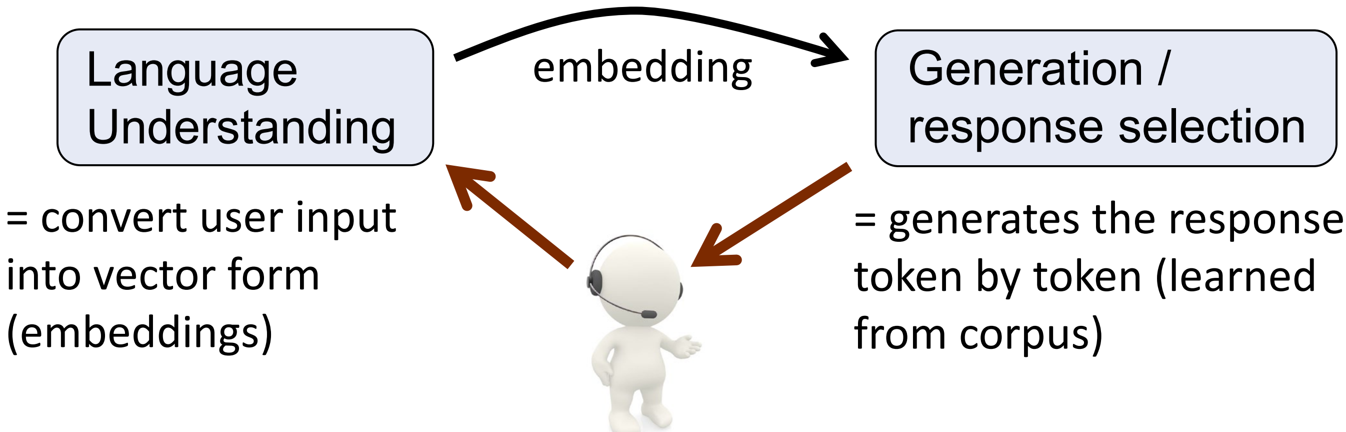
= convert user input into vector form (embeddings)

= select response from corpus that give maximum dot product

Summary

How to develop a chatbot:

- Rule-based approaches
- IR-based approaches
- **Seq-to-seq approaches**



Summary

How to develop a chatbot:

- Rule-based approaches
- IR-based approaches
- Seq-to-seq approaches
- **NLU-based approaches**

Often useful to rely on a combination of techniques – such as doing intent recognition using both rules and ML

Language Understanding

=map utterance to an intent + slots

Intent + slots

Generation / response selection

= handcrafted response or template to fill

Summary

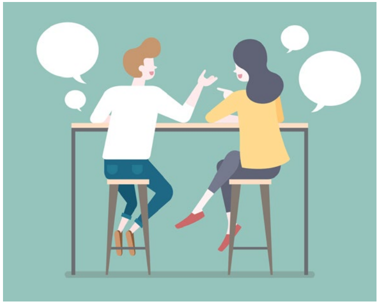
ASR: $\hat{W} = \underset{W}{\operatorname{argmax}}\, P(W|O)$

Acoustic observations
$O = o_1, o_2, o_3, \ldots, o_m$

Recognition hypothesis
$W = w_1, w_2, w_3, \ldots, w_n$

- Deep NNs have boosted ASR performance
  - But not yet a «solved problem»
  - (especially for ressource-poor languages and non-standard voices/acoustic environments)
  - *Word Error Rate metric* used for evaluation
- Disfluencies abound in spoken language

Next week



- Next week, we'll talk about *dialogue management* – that is, how do we control the flow of the interaction over time?
  - Including how to optimise dialogue policies using reinforcement learning
- And we will also talk about how to *design* and *evaluate* dialogue systems