NATURAL LANGUAGE PROCESSING

CHATBOTS

THIRUMURUGAN.R

What do you mean by a chat-bot?

Goal-oriented:

- Narrow domain
- Specific questions and tasks
- *Example*: call center
- *Model*: usually retrieval-based

Chit-chat:

- General conversation
- Human-like experience
- *Example*: entertaining bot
- *Model*: generative

Models pros and cons

Retrieval-based models

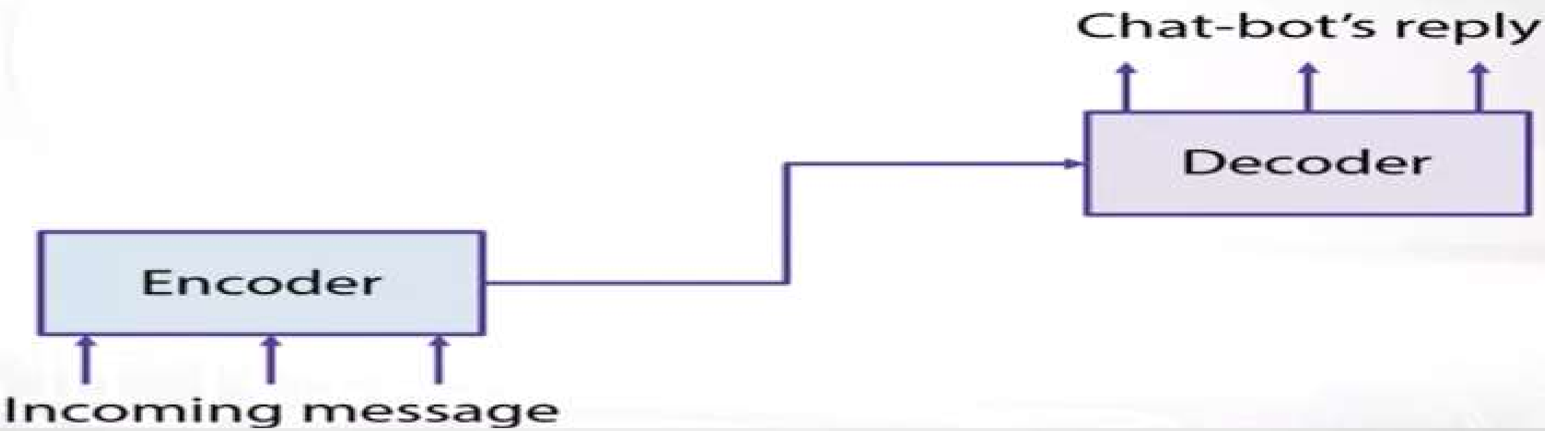
- use a repository of predefined responses
- no grammatical mistakes
- unable to handle unseen cases

Generative models

- generate new responses from scratch
- can make mistakes (especially on longer sentences)
- impression you're talking to a human

Sequence to sequence

- LSTMs for encoder and decoder
- Attention (or at least reversed input)
- Beam search for the best reply



Padding

EOS : End of sentence
PAD : Filler
GO : Start decoding
UNK : Unknown; word not in vocabulary

Q: How are you?
A: I am fine

↓

Q: [PAD,PAD,PAD,PAD,PAD,PAD,"?","you","are", "How"]
A: [GO, "I", "am", "fine", ".", EOS, PAD, PAD, PAD, PAD]

Bucketing

Bucketing solves the problem of high variance of the length of sentences by putting them into buckets of different sizes.

Buckets for the length intervals (0, 5], [5, 10], (10, 20], …

Q: How are you?

A: I am fine

↓

Q: [PAD, "?", "you", "are", "How"]

A: [GO, "I", "am", "fine", ".", EOS, PAD, PAD, PAD, PAD]

Trained on movies subtitles

Human: what is the purpose of living?
Machine: to live forever.
Human: what is the purpose of existence?
Machine: to find out what happens when we get to the planet earth.
Human: where are you now?
Machine: i'm in the middle of nowhere.

It works! It has some problems though:

- Unrealistic topics
- Too dramatic responses

Trained on calls

Machine: hi
Human: hi
Machine: could you please let me know what is the operating system you are using?
Human: **linux**
Machine: ok this is the issue with the system performance, nasser, tools.
Human: what is that ?
Machine: **is it a windows machine?**
Human: i am using linux
Machine: let me take a remote of your machine and check if issue has been pushed properly.
Human: ok

Context of the conversation

- **RNN to keep track of the intents:**

Human: hi
Machine: how may i help you today?
Human: my computer is not responding
Machine: oh ! i sorry to hear that. are you getting an error message?
Human: yes
Machine: what recent changes did you made with this computer ?
Human: i updated my computer
Machine: i see . is this on windows 7?
Human: no it is on linux
Machine: ok . are you using the affected computer right now?
Human: no

Coherent personality

- **4-layer seq2seq trained on 25 million Twitter snippets:**

Human: Where do you live now?
Machine: I live in Los Angeles.
Human: In which city do you live now?
Machine: I live in Madrid.
Human: In which country do you live now?
Machine: England, you?

- **Persona-based model:**

Human: Where are you from?
Machine: I'm from England.
Human: In which city do you live now?
Machine: I live in London.

Diversity of the responses

Always popular responses:

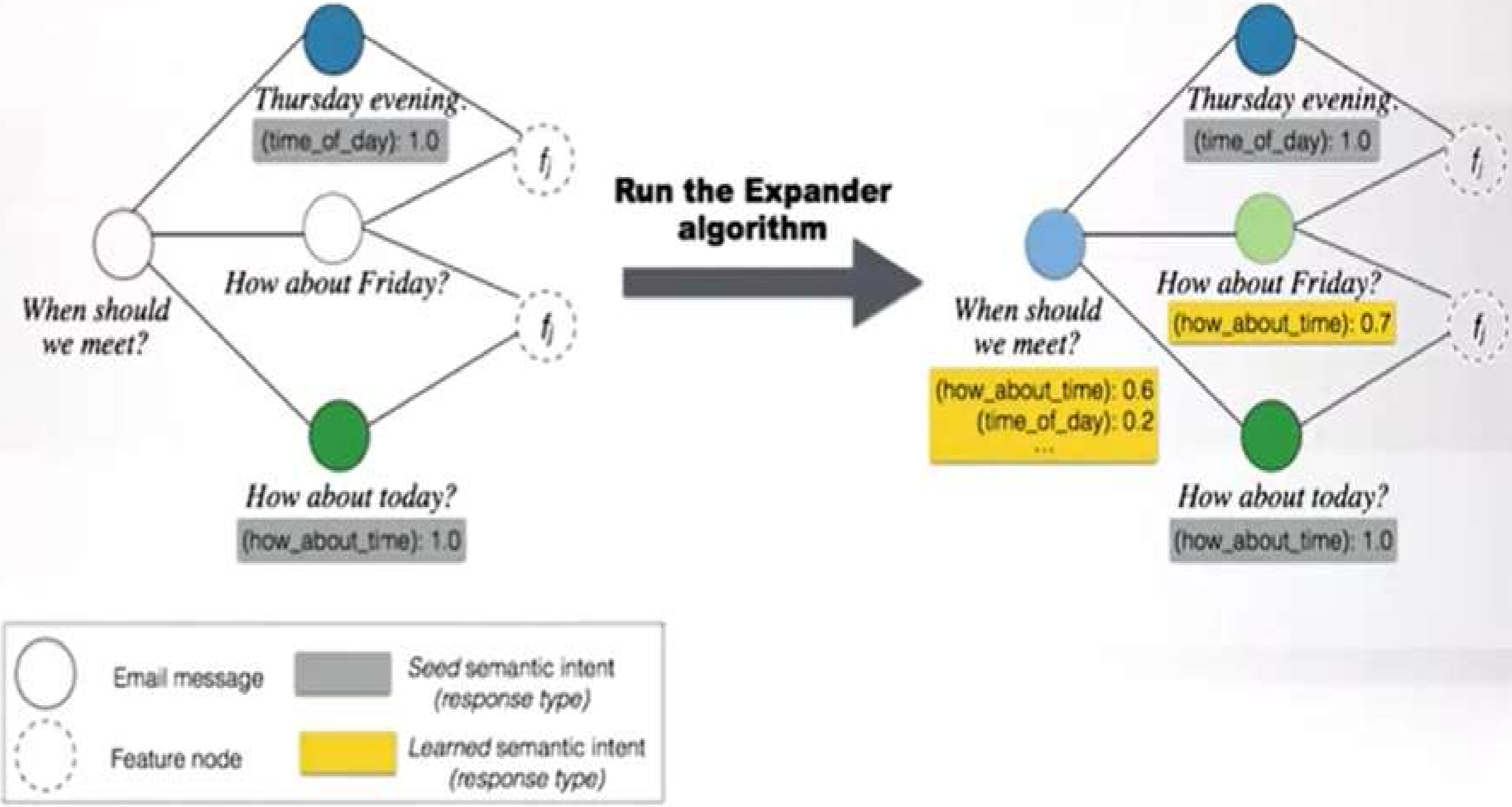
- I love you!
- I don't know

Top-3 responses (not much choice!)

- How about tomorrow?
- Wanna get together tomorrow?
- I suggest we meet tomorrow.

Kannan et. al. Smart Reply: Automated Response Suggestion for Email, KDD 2016.

Intents clustering



Google Smart Reply

Query	Top generated responses
Hi, I thought it would be great for us to sit down and chat. I am free Tuesday and Wenesday. Can you do either of those days? Thanks! –Alice	I can do Tuesday. I can do Wednesday. How about Tuesday? I can do Tuesday! I can do Tuesday. What time works for you? I can do Wednesday! I can do Tuesday or Wednesday. How about Wednesday? I can do Wednesday. What time works for you? I can do either.

Kannan et. al. Smart Reply: Automated Response Suggestion for Email, KDD 2016.

Still not a human

There are demos, and if you cherry-pick the conversation, it looks like it's having a meaningful conversation, but if you actually try it yourself, it quickly goes off the rails.

Andrew Ng

Task-oriented dialog system

You can talk to a personal assistant:

- Apple Siri
- Google Assistant
- Microsoft Cortana
- Amazon Alexa
- …

You can solve these tasks:

- Set up a reminder
- Find photos of your pet
- Find a good restaurant
- Send a message
- …

Task-oriented dialog system

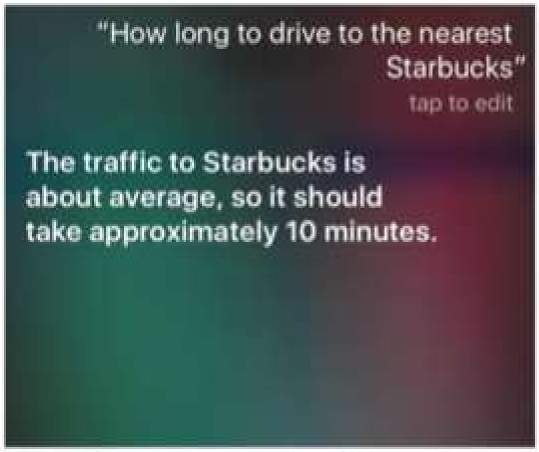
You can write to a chat bot:

- To book tickets
- To order food
- To contest parking tickets
- To track expenses
- ...

Utterance

Intent classification

- What does the user want?
- Which predefined scenario is the user trying to execute?



Intent: **nav.time.closest**

There're many intents

- And you need to classify them to give correct answers
- This is a classification task and you can measure **accuracy**

Intent: **nav.directions.closest**

And one more example

- This time assistant needs additional information and initiates dialog

Intent: **nav.directions**

Form filling approach to dialog management

- Think of an intent as a **form** that a user needs to fill in.
- Each intent has a set of fields (**slots**) that must be filled in to execute the request.

- Example: **nav.directions** intent
  - **@FROM** slot: defaults to current geolocation
  - **@TO** slot: required

- We need a **slot tagger** to extract slots from utterance.

Form filling approach to dialog management

- Think of an intent as a **form** that a user needs to fill in.
- Each intent has a set of fields (**slots**) that must be filled in to execute the request.

- Example: **nav.directions** intent
  - **@FROM** slot: defaults to current geolocation
  - **@TO** slot: required

- We need a **slot tagger** to extract slots from utterance.

Slot filling/tagging

- You **train** it as a sequence tagging task in BIO scheme
- A slot is considered to be correct if its range and type are correct
- $\textbf{Recall} = \frac{\textbf{\# correct slots found}}{\textbf{\# true slots}}$
- $\textbf{Precision} = \frac{\textbf{\# correct slots found}}{\textbf{\# found slots}}$
- You can **evaluate** slot tagger with $\mathbf{F_1} = 2 \times \frac{\textbf{Precision} \times \textbf{Recall}}{\textbf{Precision} + \textbf{Recall}}$

Form filling dialog manager (single turn)

- User: **Give me directions to San Francisco**
  - Intent classifier: nav.directions
  - Slot tagger: @TO{San Francisco}
  - Dialog manager: *all slots are filled, here's the route*
- Agent (assistant): **Here's the route**

Form filling dialog manager (multi-turn)

- User: **Give me directions from Los Angeles**
  - Intent classifier: nav.directions
  - Slot tagger: @FROM{Los Angeles}
  - Dialog manager: *required slot is missing, where to?*
- Agent (assistant): **Where do you want to go?**
- User: **San Francisco**
  - Intent classifier: nav.directions
  - Slot tagger: @TO{San Francisco}
  - Dialog manager: *okay, here's the route* ⟷ Google Maps
- Agent (assistant): **Here's the route**

- Dialog manager: *required slot is missing, where to?*

- Agent (assistant): **Where do you want to go?**
- User: **San Francisco**
  - Intent classifier: nav.directions
  - Slot tagger: @TO{San Francisco}
  - Dialog manager: *okay, here's the route*
- Agent (assistant): **Here's the route**

We need context here

Google Maps

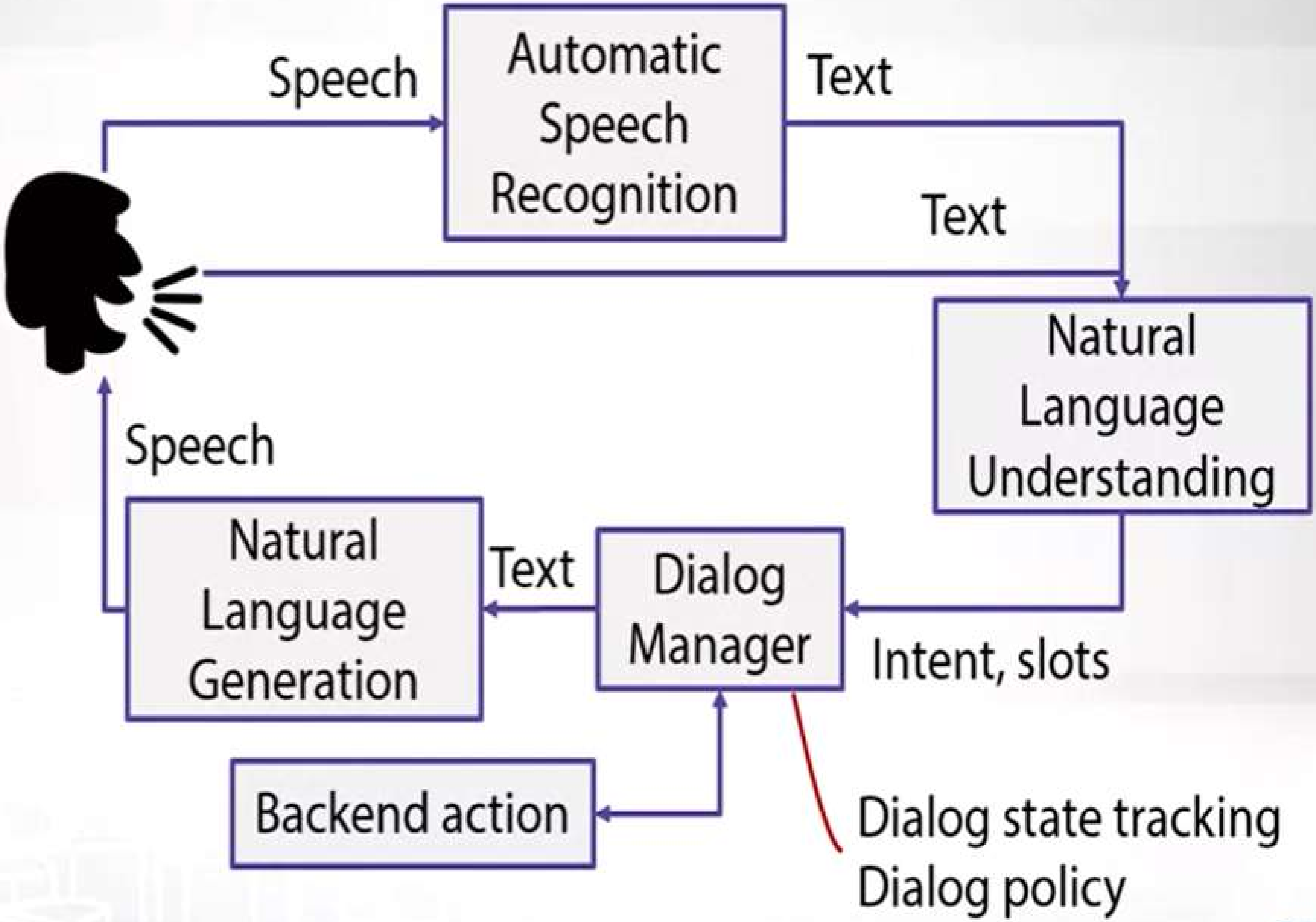
How to track context (an easy way)

- Both intent classifier and slot tagger need context (what happened before)
- Let's add simple **features** to both of them:
  - Previous utterance intent as a categorical feature
  - Slots filled in so far with binary feature for each possible slot
- Improves slot tagger F1 by 0.5%
- Reduces intent classifier error by 6.7%
- A better way: memory networks

How to track a form switch

- User: **Give me directions from Los Angeles**
  - Intent classifier: nav.directions
  - Slot tagger: @FROM{Los Angeles}
  - Dialog manager: *required slot is missing, where to?*
- Agent (assistant): **Where do you want to go?**
- User: **Forget about it, let's eat some sushi first**
  - Intent classifier: nav.find
  - Slot tagger: @CATEGORY{sushi}
  - Dialog manager: *okay, let's start a new form and find some sushi*
- Agent (assistant): **Okay, here are nearby sushi places**

Task-oriented dialog system overview



Intent classifier

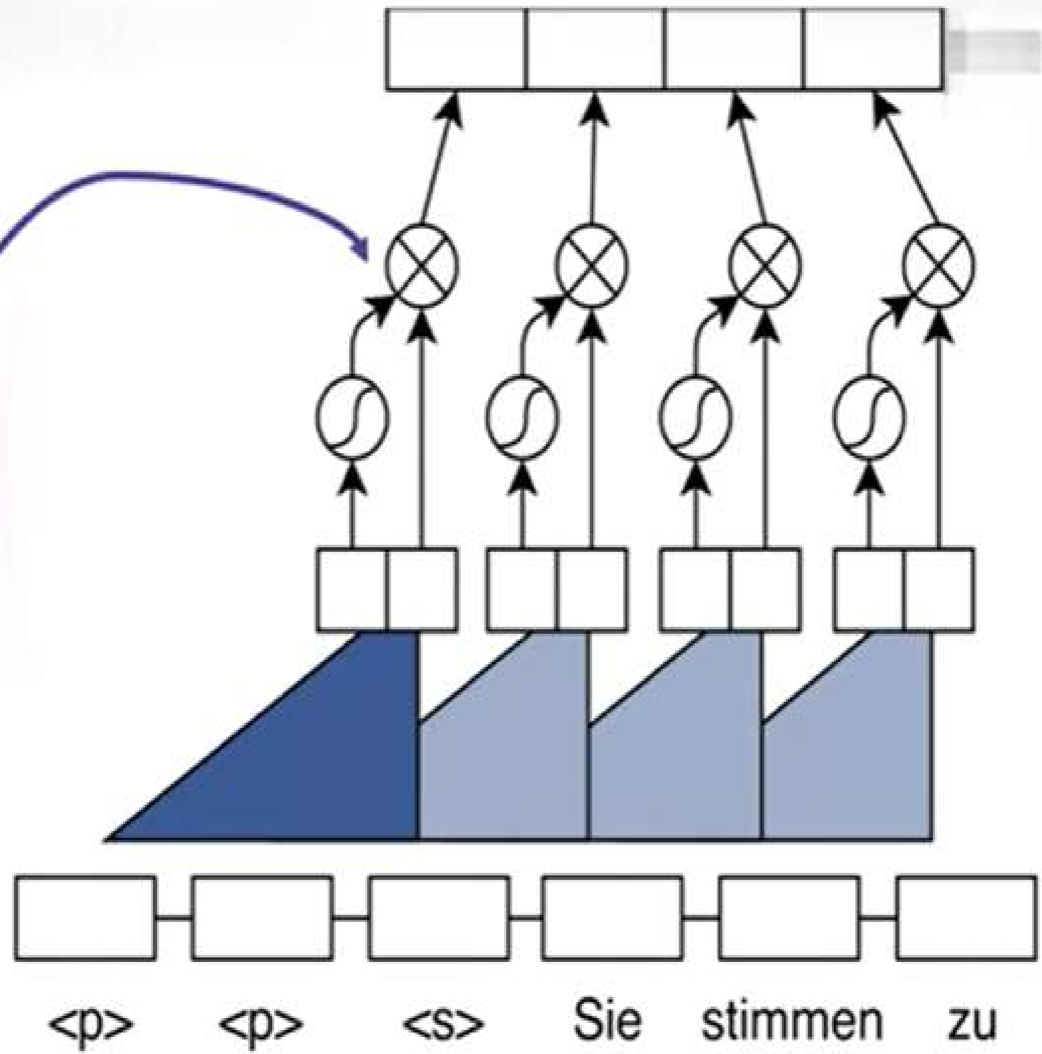
- What you can do:
  - Any model on BOW with n-grams and TF-IDF
  - RNN (LSTM, GRU, ...)
  - CNN (1D convolutions)

- CNNs can perform better on datasets where the task is essentially a key phrase recognition task as in some sentiment detection datasets.

Slot tagger

- What you can do:
  - Handcrafted rules like regular expressions
  - CRF
  - RNN seq2seq
  - **CNN seq2seq**
  - Any seq2seq with attention

CNN for sequences: Gated Linear Unit



Stacking 6 layers with kernel size 5 results in an input field of 25 elements

$$h_l(\mathbf{X}) = (\mathbf{X} * \mathbf{W} + \mathbf{b}) \otimes \sigma(\mathbf{X} * \mathbf{V} + \mathbf{c})$$

CNN for sequences: results

- They can sometimes beat LSTM in **language modeling**:

Model	Test PPL	Hardware
LSTM-1024 (Grave et al., 2016b)	48.7	1 GPU
GCNN-8	44.9	1 GPU
GCNN-14	37.2	4 GPUs

Table 3. Results for single models on the WikiText-103 dataset.

- ... and **machine translation**:

WMT'14 English-French	BLEU
Wu et al. (2016) GNMT (Word 80K)	37.90
Wu et al. (2016) GNMT (Word pieces)	38.95
Wu et al. (2016) GNMT (Word pieces) + RL	39.92
ConvS2S (BPE 40K)	40.51

https://arxiv.org/pdf/1612.08083.pdf https://arxiv.org/pdf/1705.03122.pdf

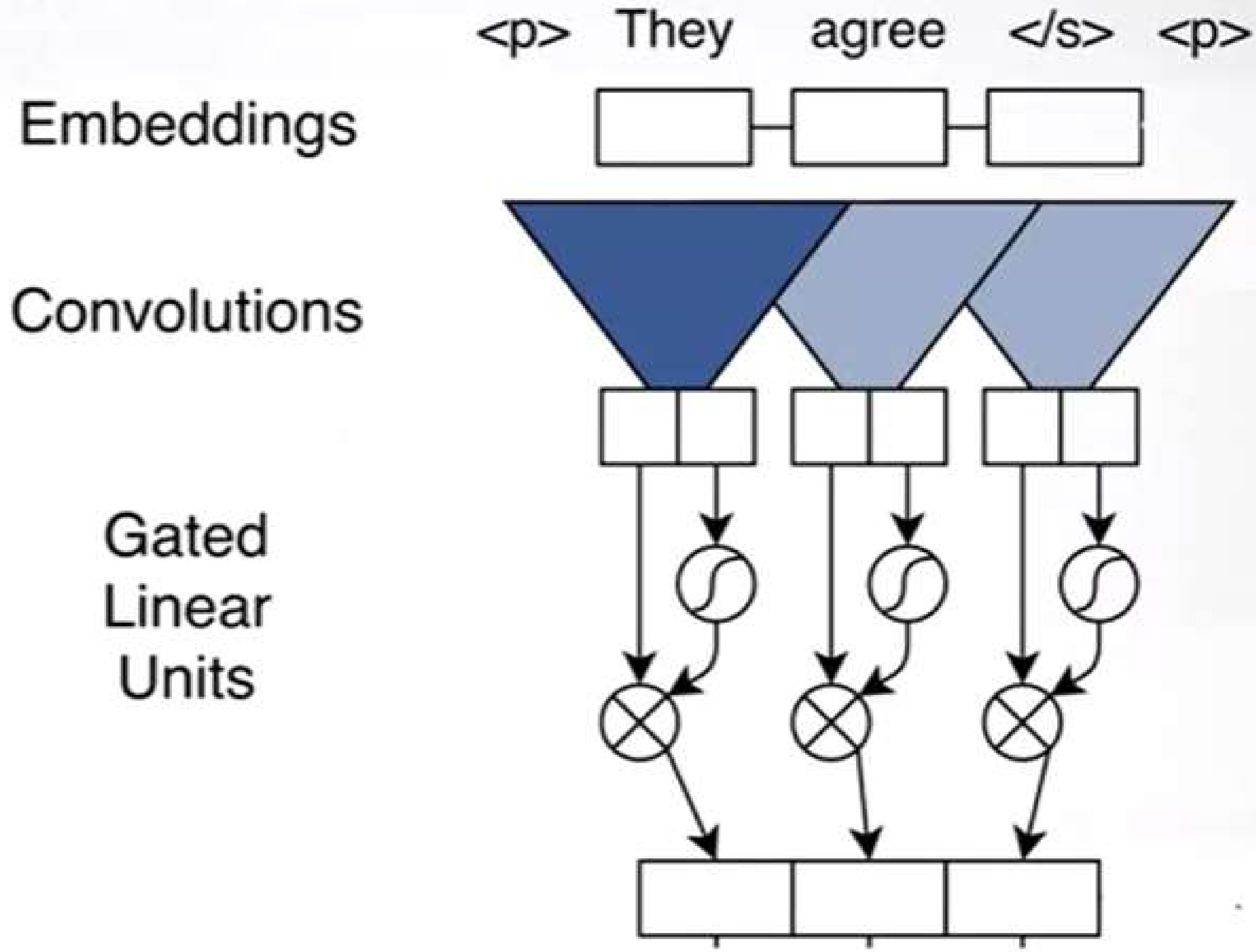
CNN for sequences: speed benefit

- They work faster than RNN:
  - During **training** we can process all time steps in parallel
  - During **testing** encoder can do the same
  - During **testing** we get higher throughput thanks to convolution optimizations in GPUs

	BLEU	Time (s)
GNMT GPU (K80)	31.20	3,028
GNMT CPU 88 cores	31.20	1,322
GNMT TPU	31.21	384
ConvS2S GPU (K40) $b = 1$	33.45	327
ConvS2S GPU (M40) $b = 1$	33.45	221
ConvS2S GPU (GTX-1080ti) $b = 1$	33.45	142
ConvS2S CPU 48 cores $b = 1$	33.45	142

Translation generation speed during testing

CNN for sequences: how encoder looks like



- Bi-directional encoder is easy
- Works in parallel for all time steps

ATIS dataset

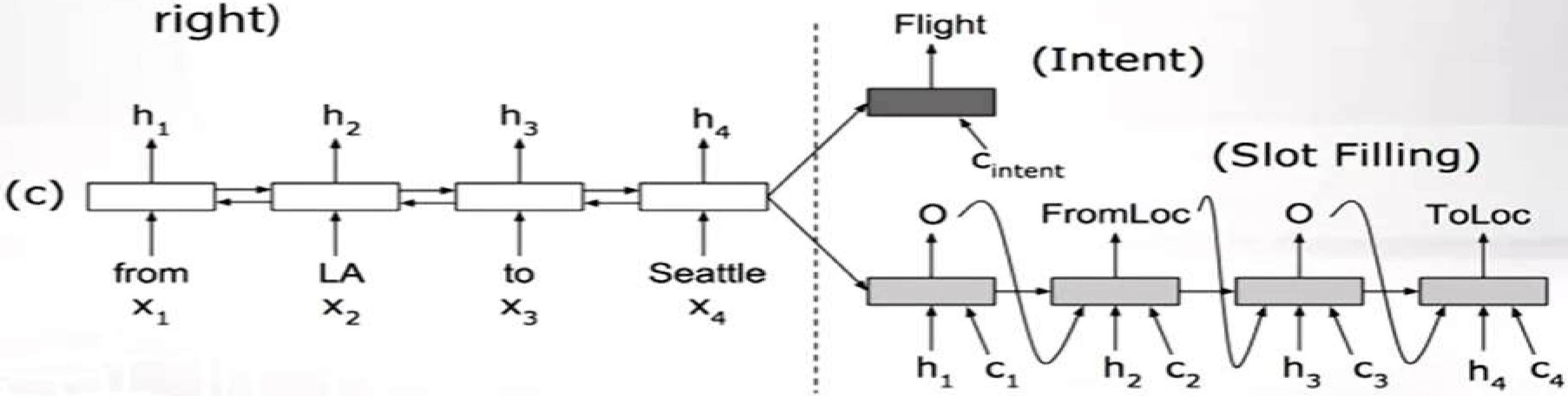
- Airline Travel Information System
- Collected in 90s
- 4978 context independent utterances
- 17 intents, 127 slot labels
- State-of-the-art: 1.79% intent error, 95.9 slots F1

Utterance	show	flights	from	Seattle	to	San	Diego	tomorrow
Slots	O	O	O	B-fromloc	O	B-toloc	I-toloc	B-depart_date
Intent	Flight							

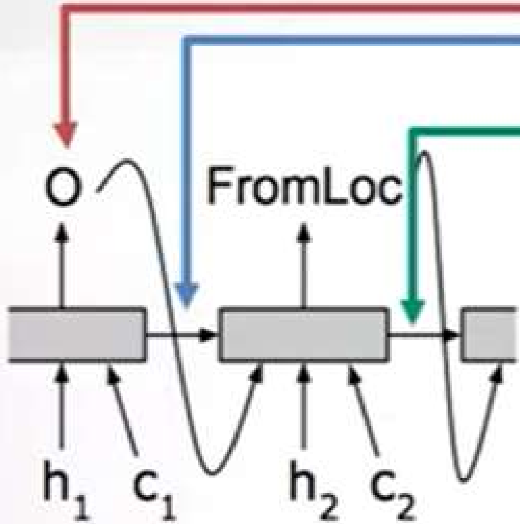
Joint training of intent classifier and slot tagger

- They both analyze the same sequence
- What if we learn representations suitable for both tasks?
- That results in more supervision and higher quality of both

- Encoder-decoder architecture for joint intent detection and slot filling
- Encoder is a bi-directional LSTM
- With aligned inputs (h_i on the right) and attention (c_i on the right)



Attention in decoder



Decoder state s_i: $s_i = f(s_{i-1}, y_{i-1}, h_i, c_i)$

Attention c_i (g is a feed forward network):

$$c_i = \sum_{j=1}^{T} \alpha_{i,j} h_j \quad \alpha_{i,j} = \frac{\exp(e_{i,j})}{\sum_{k=1}^{T} \exp(e_{i,k})}$$

$$e_{i,k} = g(s_{i-1}, h_k)$$

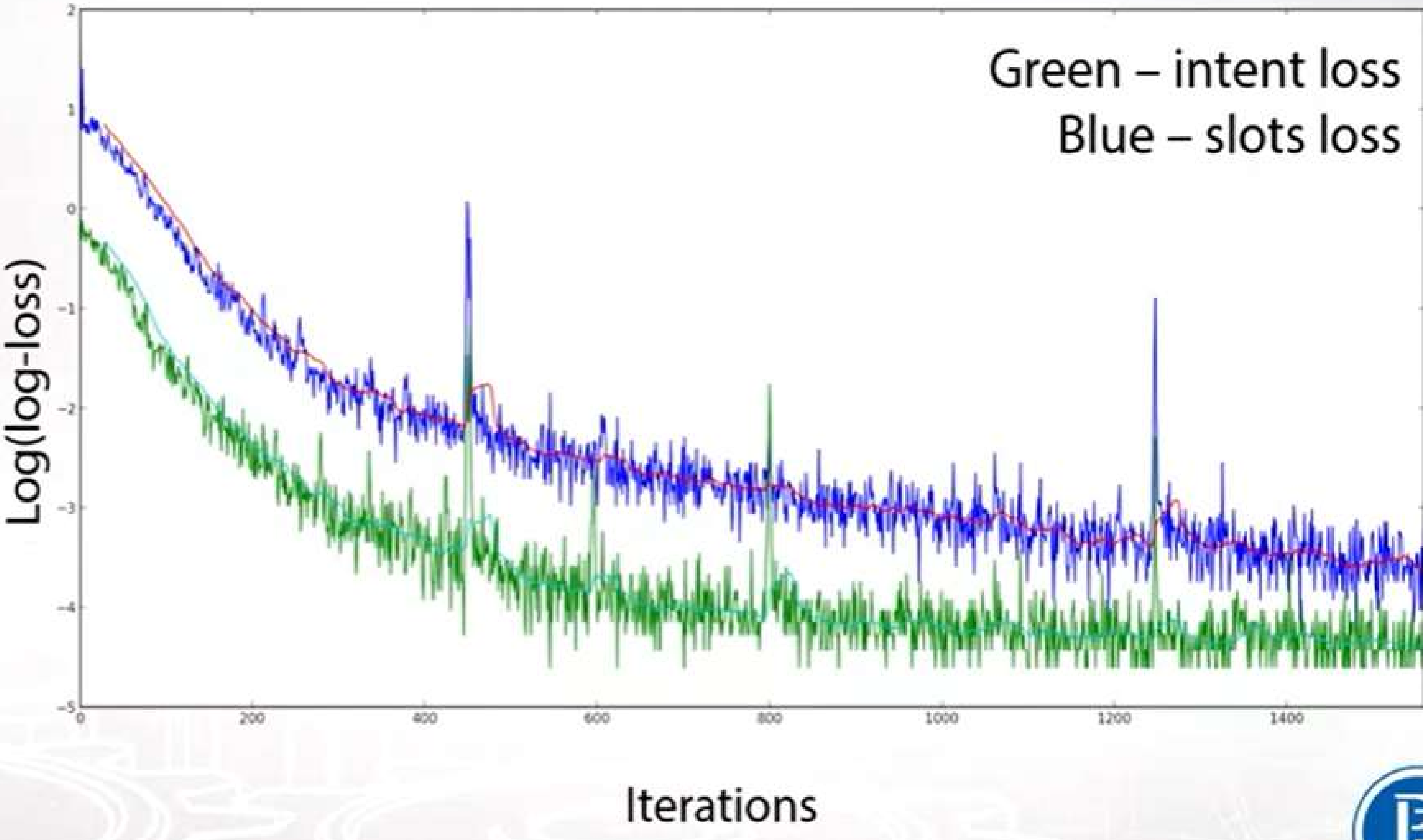
Attention weights (the darker the higher) when predicting the slot label for the last word "noon":

	O	O	B-fromloc. city_name	O	B-toloc. city_name	O	O	B-depart_time. time_relative	B-depart_time. period_of_day
$\alpha_{T,j}$	flight	from	cleveland	to	dallas	that	leves	before	noon
									T

http://www.isca-speech.org/archive/Interspeech_2016/pdfs/1352.PDF

Joint training loss

- Final training loss is a sum of losses for intent and slots



Joint training results

- Better performance on ATIS dataset:

Training	Slots F1	Intent % error
Independent training for slot filling	95.78	-
Independent training for intent detection	-	2.02
Joint training for slot filling and intent detection	**95.87**	**1.57**

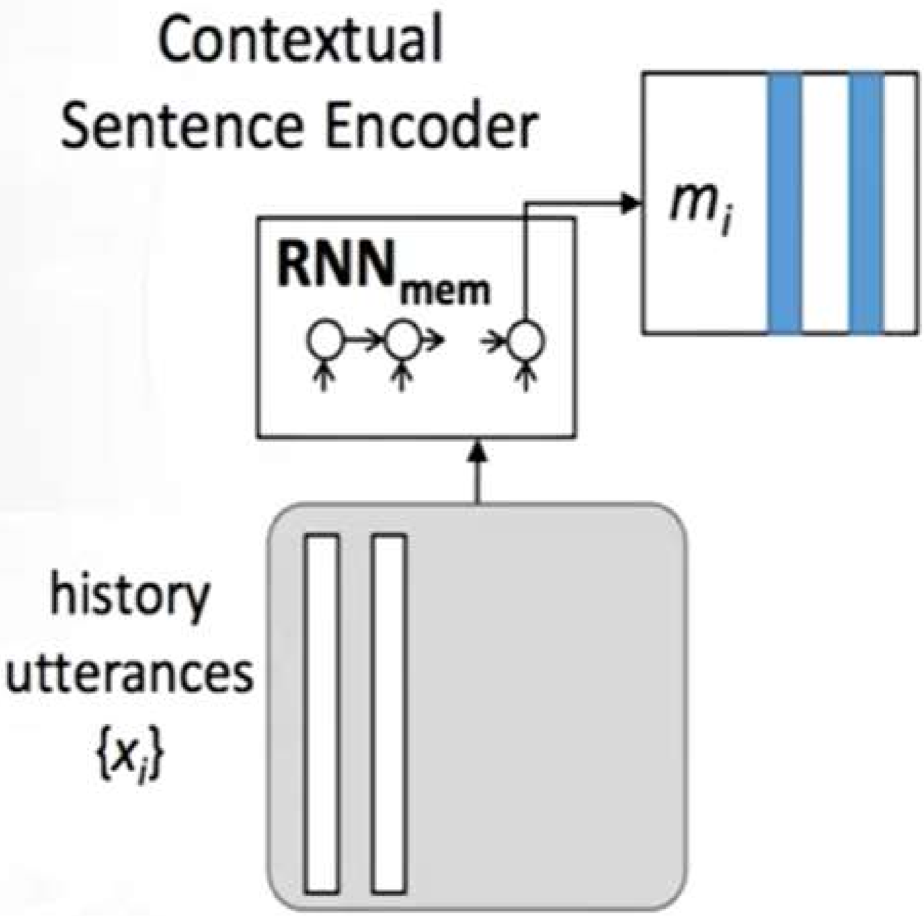
- Works faster than two separate models

Summary

- We've overviewed different options for intent classifier and slot tagger training
- People start to use CNN for sequence modeling and sometimes get better results than with RNN
- Joint training can be beneficial in terms of speed and performance
- In the next video we'll take a look at context utilization in our NLU (intent classifier and slot tagger)

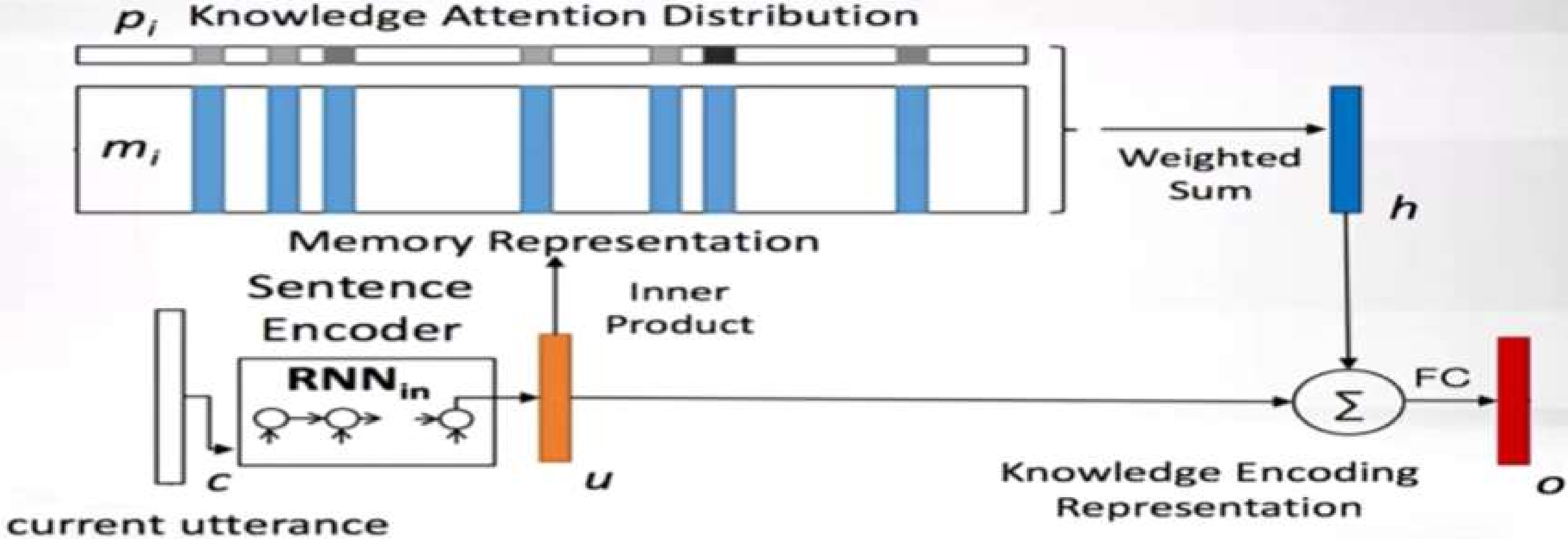
Adding context to NLU:

Let's store all previous utterances in "memory"



$$m_i = \mathrm{RNN}_{\mathrm{mem}}(x_i)$$

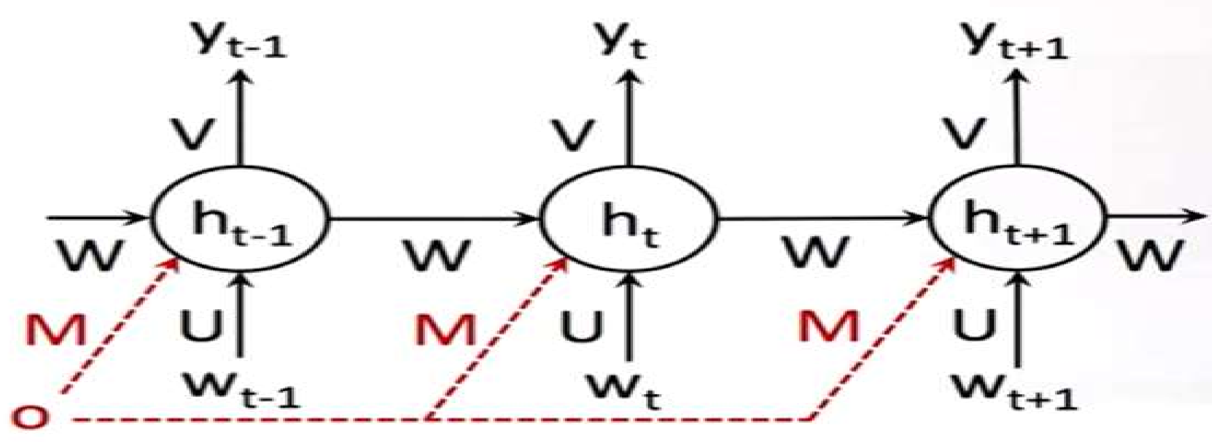
What knowledge is relevant to new utterance?



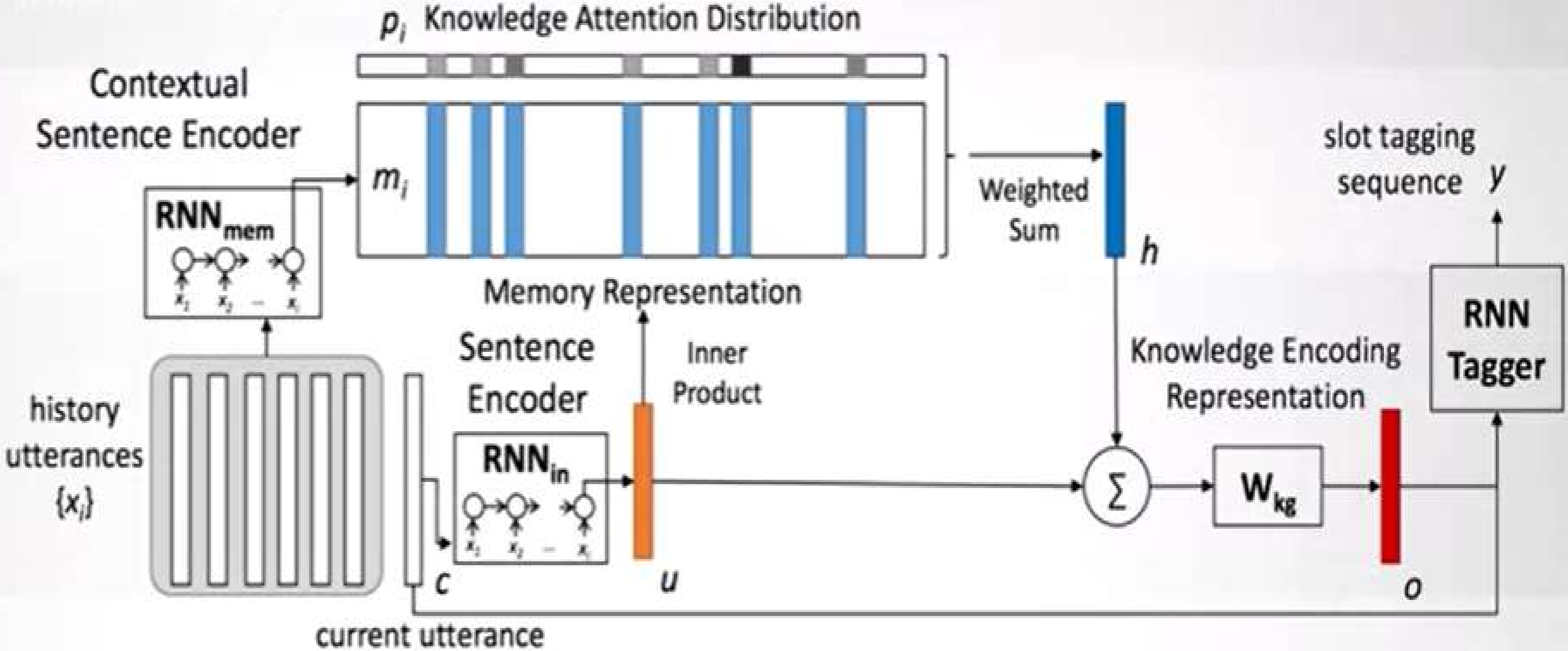
$$u = \text{RNN}_{\text{in}}(c)$$

Tagging current utterance with knowledge

- We add knowledge representation in final RNN tagger:



How to track context (with memory networks)



Chen et al, 2016

- We encode previous utterances to store them in **"memory"** as dense vectors
- We use **attention** mechanism to retrieve relevant prior knowledge about the conversation

How to track context (with memory networks)

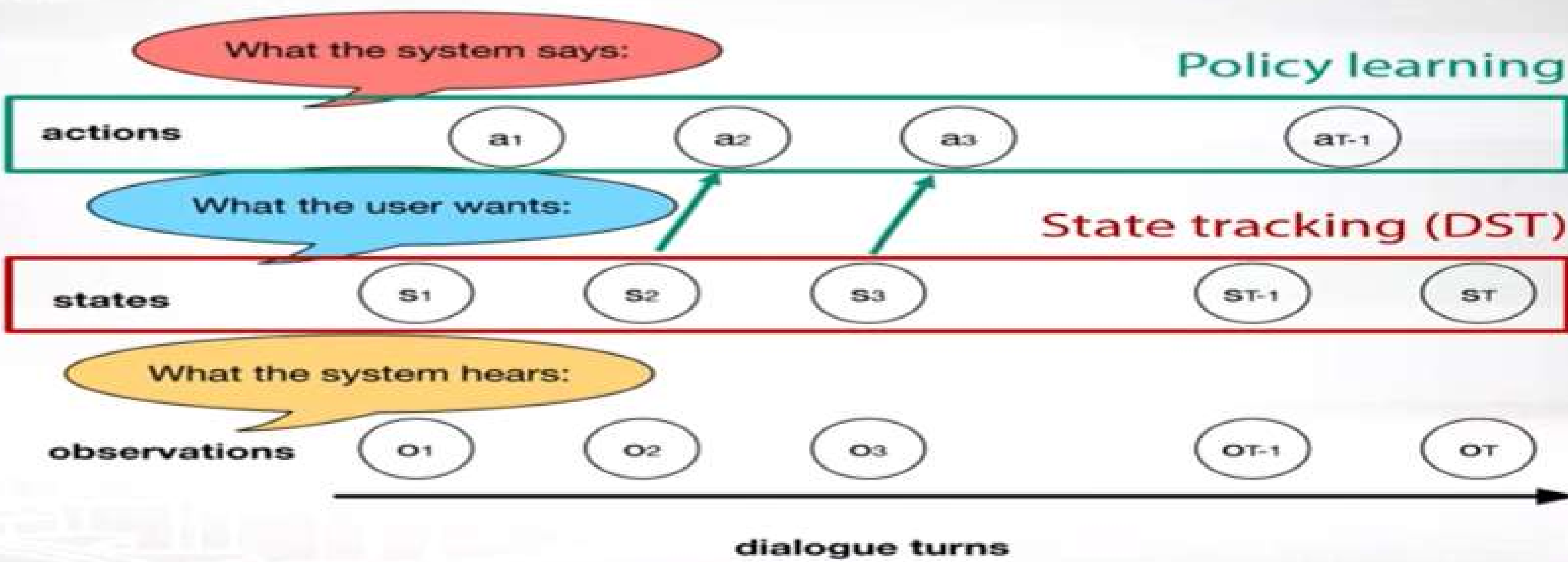
- Evaluation results for slot tagger:
  - Multi-turn dataset
  - F1-measure

Model	First turn	Other turns	Overall
RNN tagger wo context	55.8	45.7	47.4
Memory Network	73.2	65.7	67.1

Dialog manager

- **State tracker** (requires hand-crafted states)
  - Queries the external database or knowledge base
  - Tracks the evolving state of the dialog
  - Constructs the state estimation
- **Policy learner**
  - Takes the state estimation as input and chooses a dialog action

State tracking and policy learning



DSTC 2 dataset

- Dialog State Tracking Challenge, collected in 2013
- **Human-computer** dialogs (finding a restaurant in Cambridge)
  - 3324 telephone-based dialogs, people were recruited using Amazon Mechanical Turk
  - *Dialog systems used: an MDP / POMDP for tracking the dialog state, and a hand-crafted policy / policy learnt using reinforcement learning*
- Labeling procedure:
  - Utterances transcription using Amazon Mechanical Turk
  - Annotation by heuristics
  - Checked & corrected by hand

DSTC 2 dataset

- **Dialog state:**
  - ***Goals***: A distribution over the values of each informable slot
  - ***Method***: A distribution over methods: by name, by constraints, by alternatives or finished
  - ***Requested slots***: A probability for each requestable slot that it has been requested by the user and the system should inform it
- User dialog acts: inform, request, negate, confirm, …
  - What part of town is it? → `request(area)`
- Method is inferred from act and goals:
  - `inform(food=chinese)` → "by constraints"

DSTC 2 dialog excerpt

Utterance	I'm looking for an expensive restaurant with venetian food
Goals	food=venetian, pricerange=expensive
Method	byconstraints
Requested slots	[]

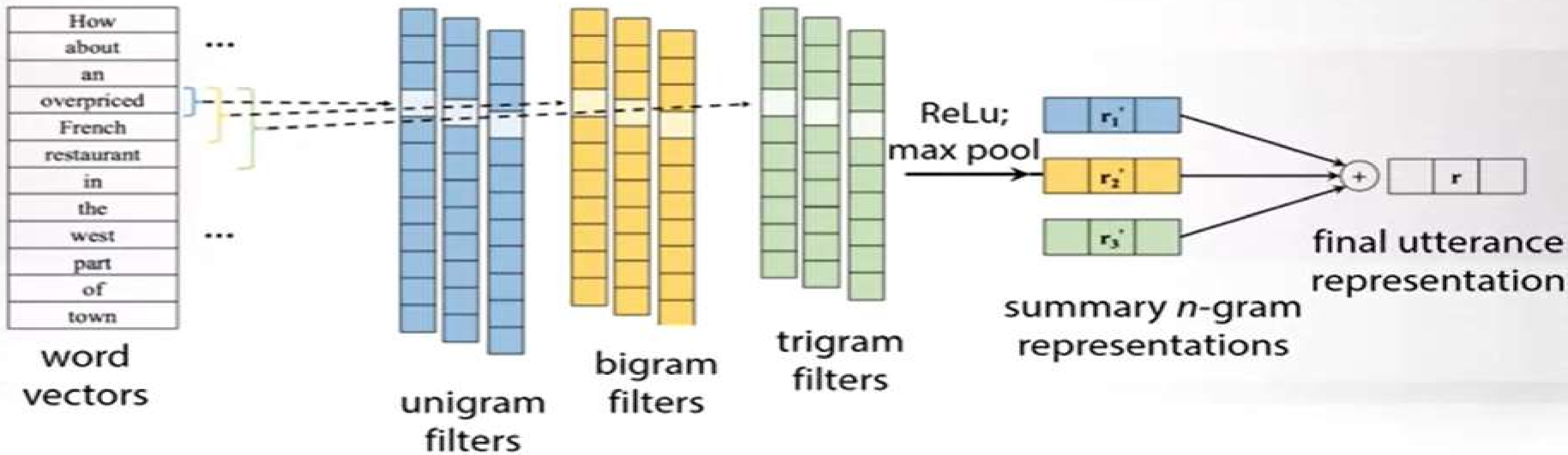
Utterance	Is there one with thai food?
Goals	**food=thai**, pricerange=expensive
Method	byconstraints
Requested slots	[]

Utterance	Can I have the address?
Goals	food=thai, pricerange=expensive
Method	byconstraints
Requested slots	[**addr**]

DSTC 2 results

- Best results after competition:
  - Goals: 65% correct *combinations*
  - Method: 97% correct
  - Requested slots: 95% correct

Utterance representation



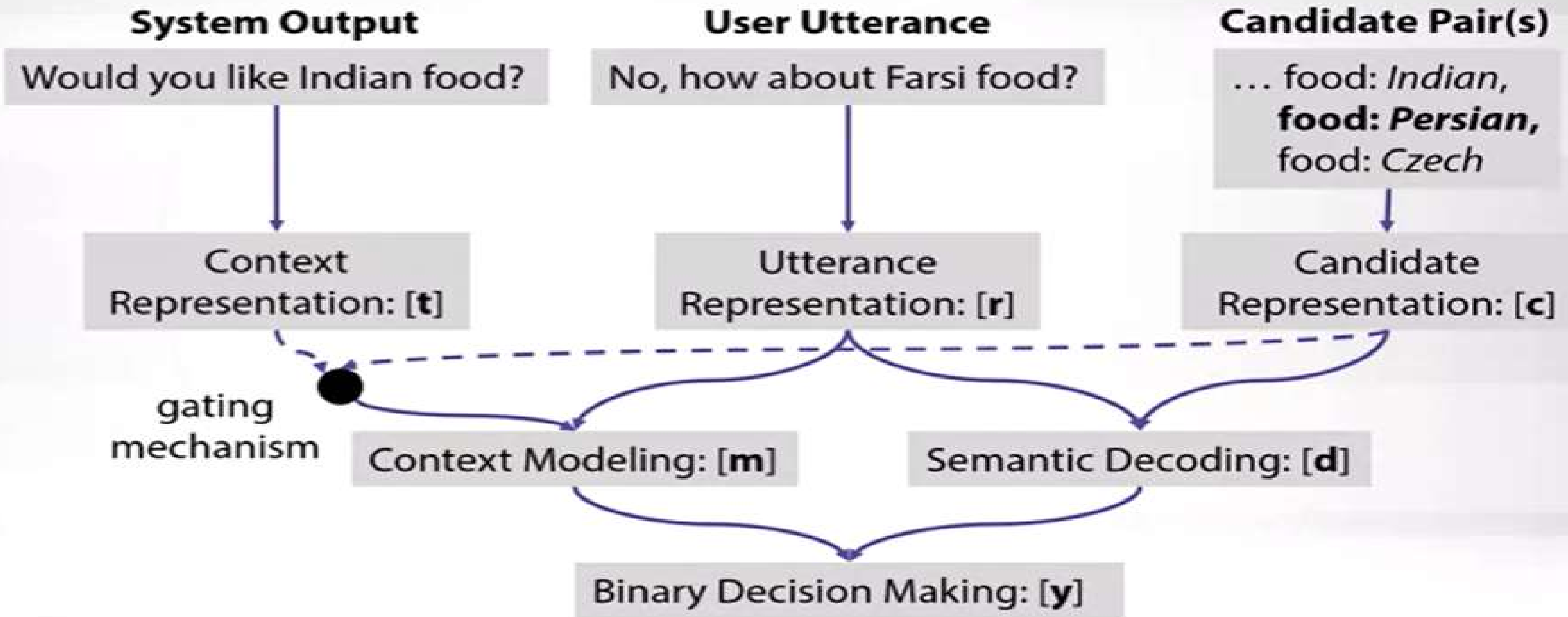
Neural Belief Tracker results

DST Model	DSTC2		WOZ 2.0	
	Goals	Requests	Goals	Requests
Delexicalisation-based Model	69.1	95.7	70.8	87.1
Delexicalisation-based Model + Semantic Dictionary	72.9*	95.7	83.7*	87.6
Neural Belief Tracker: NBT-DNN	72.6*	96.4	**84.4***	91.2*
Neural Belief Tracker: NBT-CNN	**73.4***	**96.5**	84.2*	**91.6***

Rule-based state tracking

- Train a good NLU (intents and slots)
- Make simple hand-crafted rules for dialog state change

Neural Belief Tracker



Joint NLU/DST

Frames dataset

- Collected in 2016
- **Human-human** goal-oriented dataset
  - 12 participants, 20 days, 1369 dialogues
  - Two humans talked to each other via a Slack chat

Find a vacation between September 1st and September 8th to Havana from Stuttgart for under $700. Dates are not flexible. If not available, then end the conversation.

User

You have access to a database of 250+ packages, each composed of a hotel and round-trip flights. Provide help via a chat interface.

Wizard

Frames dataset

- It introduces a new task called **frame tracking**, which extends state tracking to a setting where several states are tracked simultaneously
- In this dataset users can compare results corresponding to different constraints and go back-and-forth between results

Author	Utterance	Frame
User	I'd like to book a trip to Atlantis from Caprica on Saturday, August 13, 2016 for 8 adults. I have a tight budget of 1700.	1
Wizard	Hi...I checked a few options for you, and unfortunately, we do not currently have any trips that meet this criteria. Would you like to book an alternate travel option?	1
User	Yes, how about going to Neverland from Caprica on August 13, 2016 for 5 adults. For this trip, my budget would be 1900.	2
Wizard	I checked the availability for those dates and there were no trips available. Would you like to select some alternate dates?	2

Dialogue excerpt with active frame annotation

- It's annotated with the following:
  - Dialogue acts, slot types, slot values, and references to other frames for each utterance.
  - The ID of the currently active frame.
- Examples:

`inform(category=2.5)`
2.5 stars will do

`offer(ref=[6], seat=business, price=1002.27)`
What about a $1002.27 business class ticket to San Francisco?

Why do we want to utilize lexicon?

- Let's take ATIS dataset
- It has finite set of cities in training
- Will the model work for a new city?
- We have **a list of all cities**, why not use it?

- Another example
- Imagine you need to fill a slot "music artist"
- We have **all music artists** in the database like musicbrainz.org
- How can we use it?

Let's add lexicon features to input words

- Let's **match every n-gram** of input text against entries in our lexicon

Take me to San Francisco

- A match is successful when **the n-gram matches the prefix or postfix** of an entry and is at least half the length of the entry

Matches:

"San" → "San Antonio"
"San" → "San Francisco"
"San Francisco" → "San Francisco"

- When there are multiple **overlapping matches**:
  - Prefer **exact** matches over partial
  - Prefer **longer** matches over shorter
  - Prefer **earlier** matches in the sentence over later

Matches encoding

We will use **BIOES** coding (Begin, Inside, Outside, End, Single)

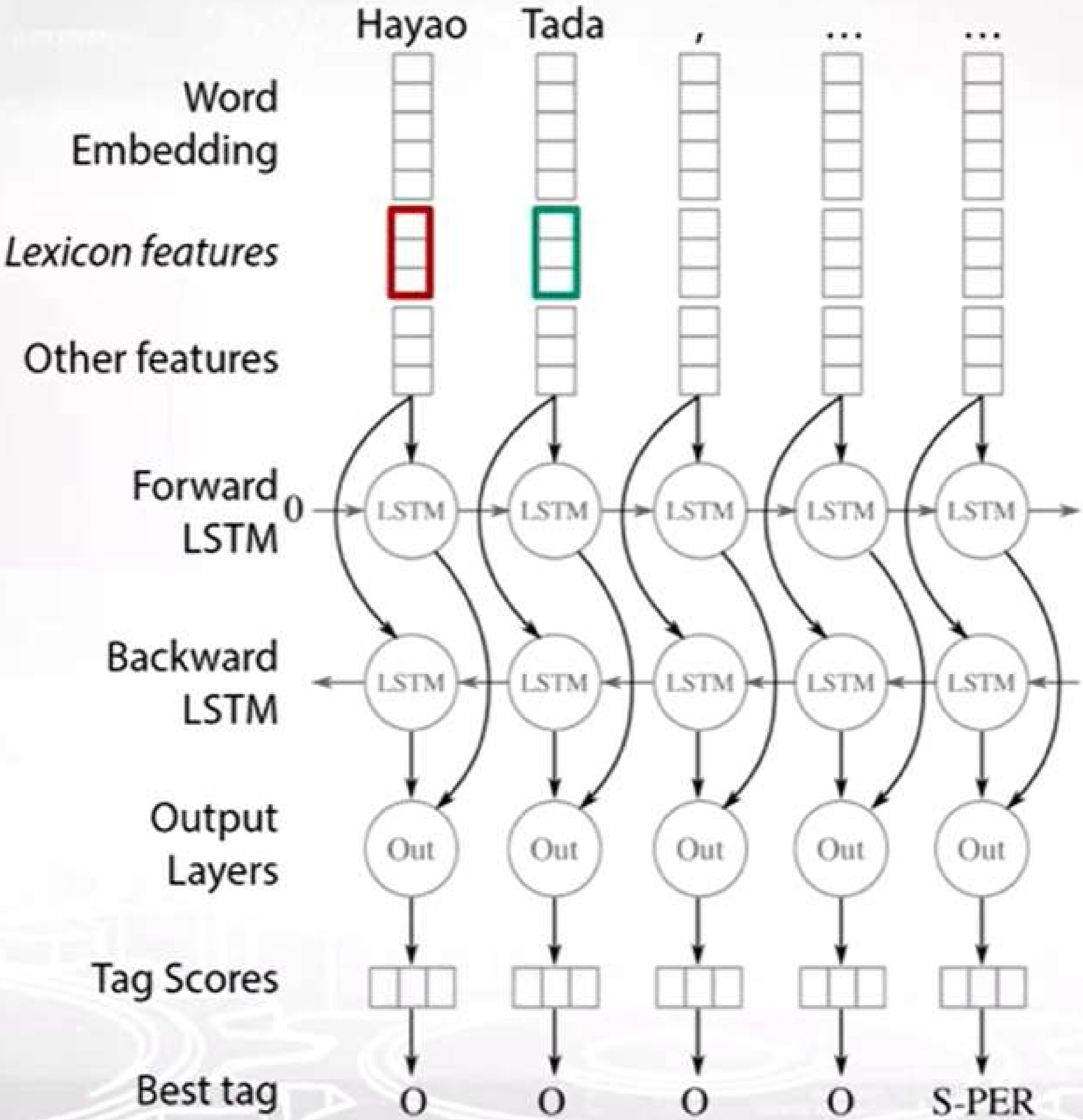
- B – if token matches the beginning of some entity
- B, I – if two tokens match as prefix
- I, E – if two tokens match as postfix
- S – if matched single token entity
- …

Example for 4 lexicon dictionaries:

Text	Hayao	Tada	,	commander	of	the	Japanese	North	China	Area	Army
LOC	–	–	–	–	–	B	I	–	S	–	–
MISC	–	–	–	S	B	B	I	S	S	S	S
ORG	–	–	–	–	–	B	I	$\boxed{B}$	$\boxed{I}$	$\boxed{I}$	$\boxed{E}$
PERS	$\boxed{B}$	$\boxed{E}$	–	–	–	–	–	–	S	–	–

B, I, O, E, S are later encoded as one-hot vectors

Adding these features to our model



These're our lexicon features

Text	Hayao	Tada	,
LOC	–	–	–
MISC	–	–	–
ORG	–	–	–
PERS	B	E	–

https://arxiv.org/pdf/1511.08308v4.pdf

Training details

- You can **sample** your **lexicon** dictionaries so that your model learns the context of entities as well as lexicon features
- This procedure helps **to detect unknown entities at testing**

- You can **augment** your dataset replacing slot values with values from the same lexicon:

Take me to **San Francisco**

↓

Take me to **Washington**

Dialog policy

- Dialog state → Agent act

- Policy execution examples:

`inform(location="780 Market St")`
The nearest one is at 780 Market St

`request(location)`
What is the delivery address?

Simple approach: hand crafted rules

- You have NLU and state tracker
- You can come up with hand crafted rules for policy

Optimizing dialog policies with ML

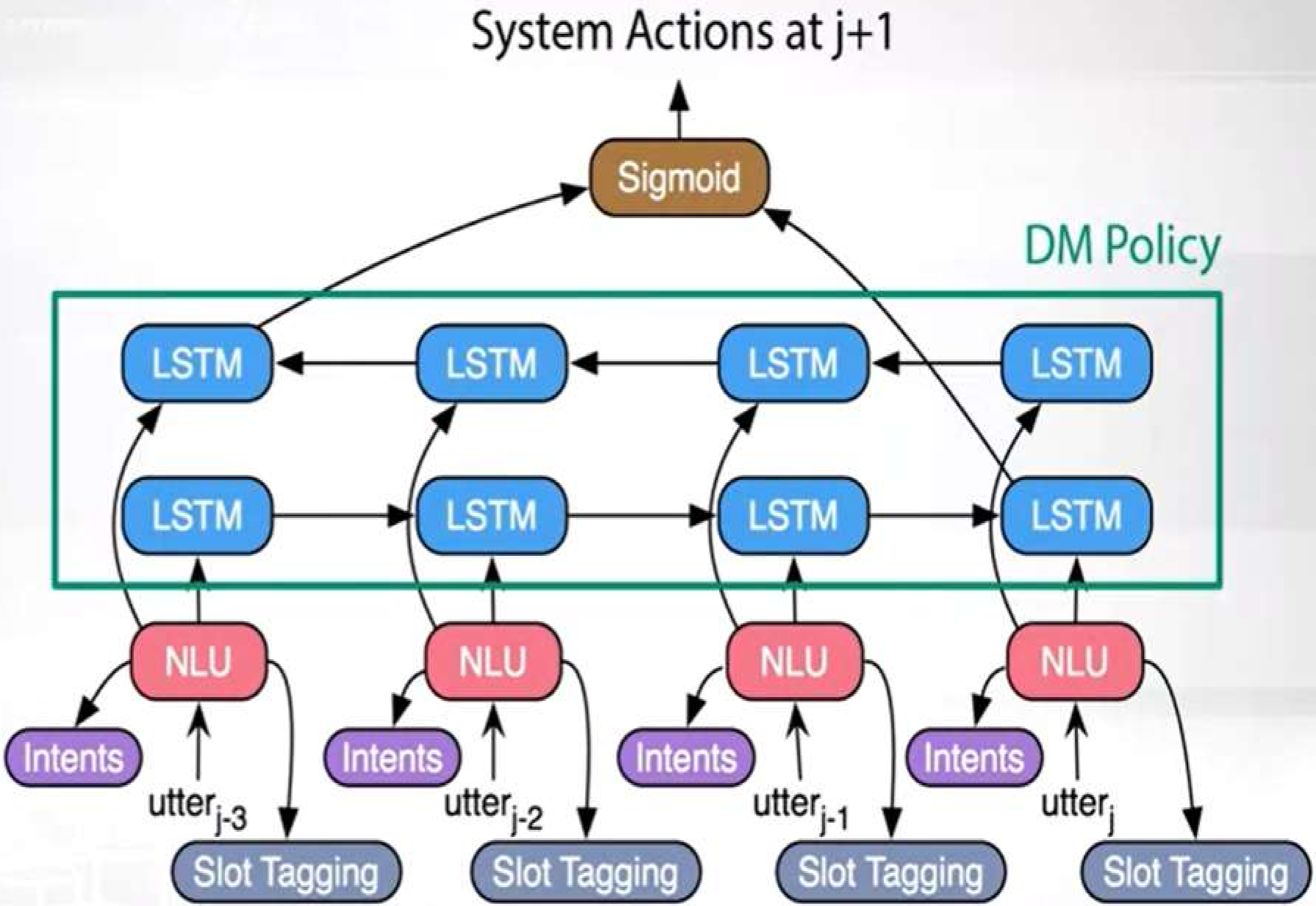
Supervised learning:

- You train to imitate the observed actions of an expert
- Often requires a large amount of expert-labeled data
- Even with a large amount of training data, parts of the dialogue state space may not be well-covered in the training data

Reinforcement learning:

- Given only a reward signal, the agent can optimize a dialogue policy through interaction with users.
- RL can require many samples from an environment, making learning from scratch with real users impractical
- That's why we need *simulated users* for RL

Joint NLU and DM



Joint NLU and DM results

Model	DM	NLU
Baseline (CRF + SVMs)	7.7	33.1
Pipeline-BLSTM	12.0	36.4
Joint Model	**22.8**	**37.4**

Frame level accuracies
(it counts only when the whole frame parse is correct)

Evaluation

- **NLU:**
  - Turn-level metrics: intent accuracy, slots F1
- **DM:**
  - Turn-level metrics: state tracking accuracy, …
  - Dialog-level metrics: task success rate, reward, …