

Quantized Dialog

David Nahamoo

In Collaboration with

USA{Chulaka Gunasekara, Lazaros Polymenakos, David Echeverria Ciaurri,
Jatin Ganhotra, Kshitij Fadnis, Siva Sankalp Patel},
India{Dinesh Raghu, Sachindra Joshi, Vineet Kumar, Dhiraj Madan}

IBM Research

Conversational AI Workshop, Long Beach, Dec 8 2017

What is a Conversation?

To most of us.... A conversation is a form of interactive communication, conducted typically through multiple exchanges

Human: *what is integrity ?*

Machine: *i'm sorry, i don't know what else to say ?*

Human: *be moral !*

Machine: *be a man !*

Human: *I really like our discussion on morality and ethics...*

Machine: *and how I'm not in the mood for a philosophical debate*

Human: *what do you like to talk about ?*

Machine: *nothing .*

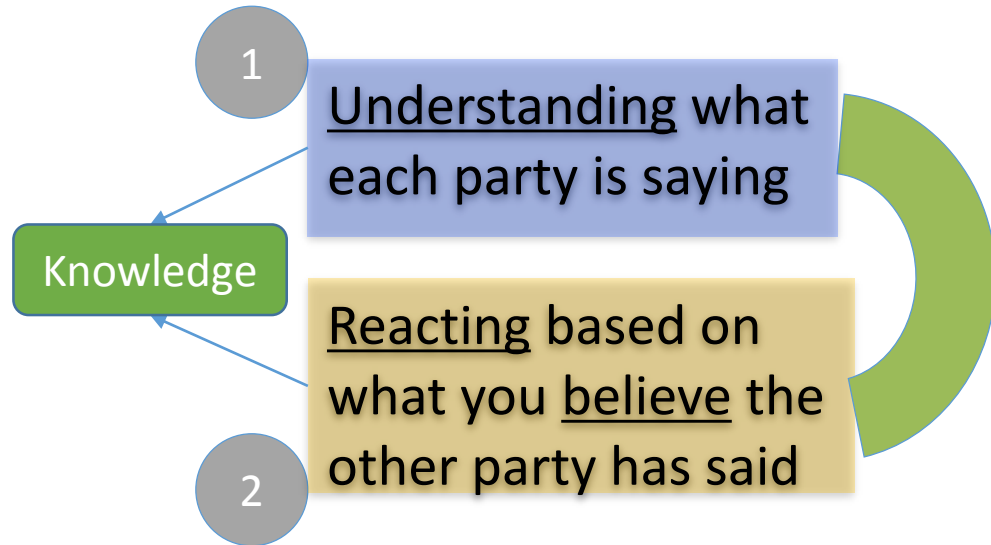


No generally accepted definition of conversation exists:

- *involves at least two people*
- *oral exchange of sentiments, observations, opinions, or ideas*
- *“informal” exchange of thoughts, information, ...*

Engineering of a Conversation Machine

Irrespective of “What **Conversation** is or isn’t, the act of “**Conversing**” requires:



- Understanding requires Language Knowledge
- Believing requires Task Knowledge
- Reacting requires Logic applied to Task Knowledge
- Reacting also needs Language Knowledge to respond

Interaction Dimensionality is extremely large

Vocabulary: 100K, Max Length: 20, Max # of Turns: 10

*# of possibilities in the Hypothesis Search Space: (100K) ** 200*

User Dimensionality is very large but less clear

Situational Context: Age, Location, Time, ...

Affective Context: Emotion, Mood, Personality

1. Engineering of a Conversation Machine sure seems very daunting
2. In Traditional Computing, raw space is even larger. Rules, Structures, ..., have made it possible
3. What are the appropriate patterns for Engineering Conversational Computing?

Building a Conversational System

Far from self-organizing design that learns thru guidance, observation, & experimentation

- **Explicit** – Designer explicitly annotates all aspects of language and task knowledge and programs the language and flow processing
- **Implicit** – Designer preserves the structure, semi-structure, & unstructured nature of knowledge in the form of a Task Semantic Representation and uses inference engines to implement the flow processing. Language processing remains explicit
- **End2End** – No explicit or implicit notion of task knowledge by the designer. Flow processing is directly derived from the utterances which is embedded in the language processing
- **Hybrid** – A fused combination of Explicit, Implicit, and End2End approaches are used

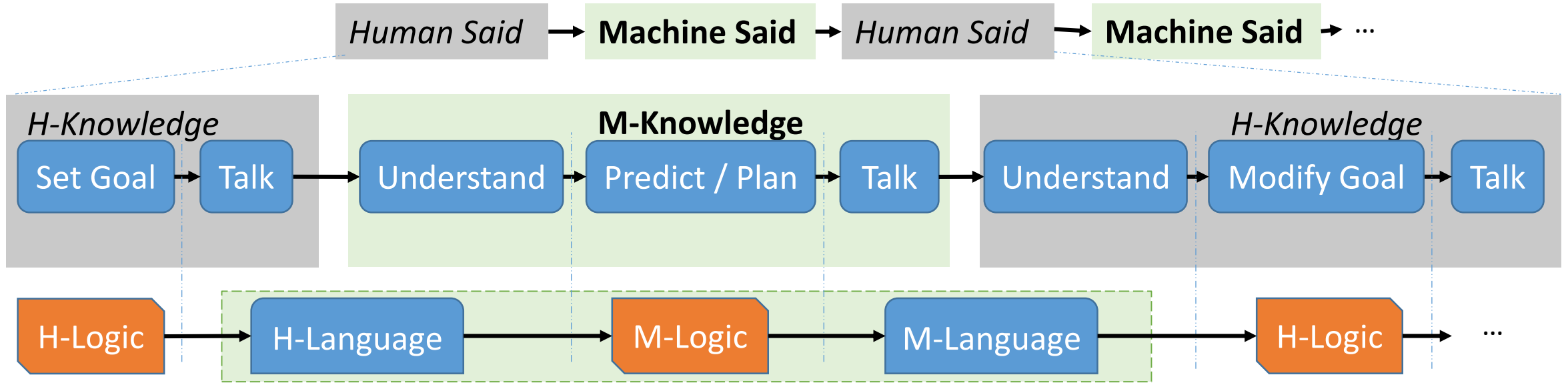
End2End Conversational Systems

Problem Statement:

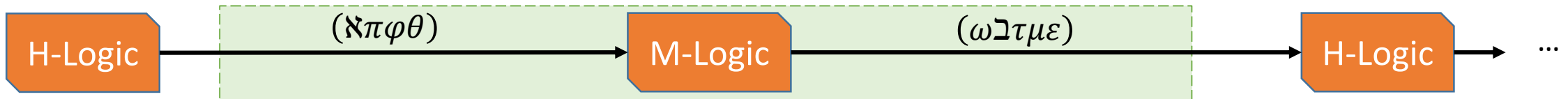
Given $U_{trn}(i, 1)$, $U_{trn}(i, 2)$, ... , $U_{trn}(i, n_i)$ training utterances for conversations $i = 1, \dots, I$
Build a conversational system that can accurately predict the next utterance $U_{tst}(j + 1)$ having observed the conversation context $U_{tst}(1)$, $U_{tst}(2)$, ... , $U_{tst}(j)$

- **Generation** – Use a Seq2Seq decoder such as HRED with each context utterance represented as a sequence of word embedding vectors and the output utterance is decoded one word at a time
- **Retrieval** – The output utterance is optimally selected from all the “training” utterances
- **Model Based Retrieval (or G): Separate Logic from Language**
 - We have a lot of experience in Language processing
 - We are at the beginning of Neuromorphic Logic processing
 - Separation of logic from language should help: Complexity Reduction, Product Space, ...

Language and Logic: Conversational Systems Processing



- Model each utterance in a conversation as a carrier-language Modulated by an embedded-logic
- Demodulate the carrier-language in order to model the logic sequence in a conversation

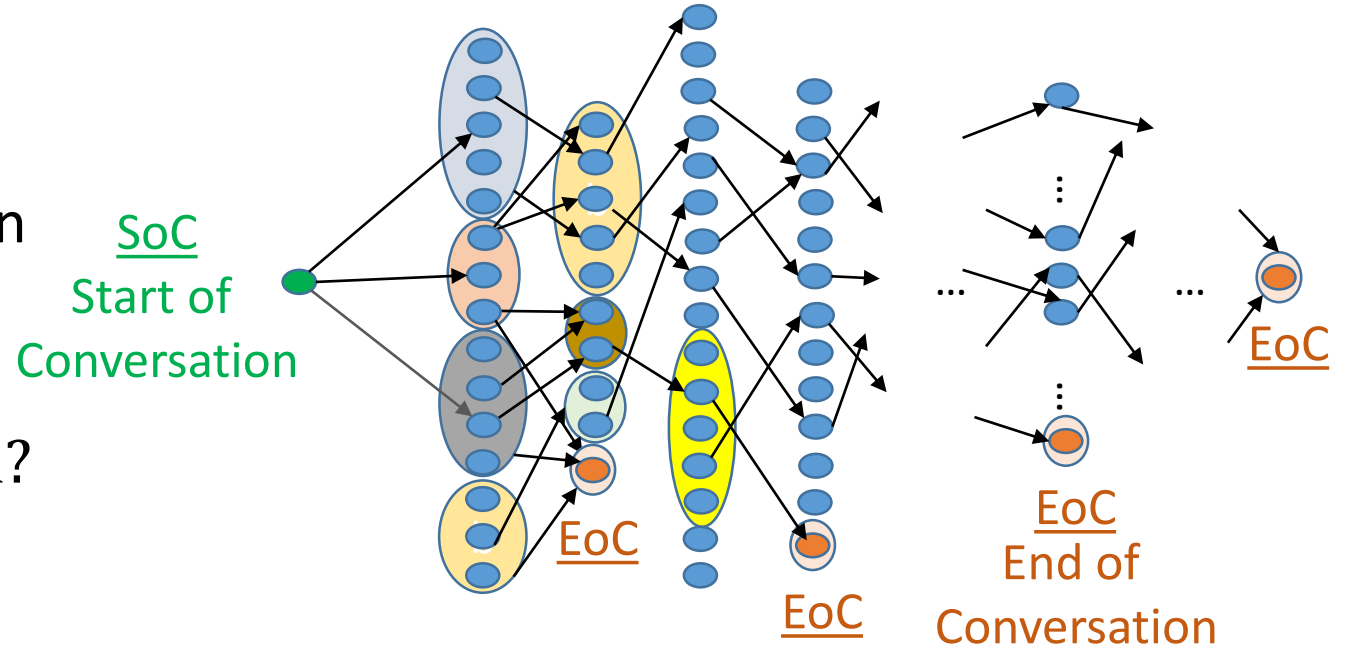


- Modulate the carrier language back to map predicted logic position into surface language form

Quantized Dialog Formulation

Demodulate the carrier language by quantizing the conversational utterances into a small number of clusters

- Let $\vec{U}_n = U_1, U_2, \dots, U_n$ be the utterances in a conversation between two speaker
- What is $P(U_n | \vec{U}_{n-1})$ in a QD Model?



Assume that clustering has achieved ultimate separation between Logic & Language

- $P(C(U_n) | \vec{U}_{n-1}) = P(C(U_n) | C(U_1), C(U_2), \dots, C(U_{n-1}))$
- $P(U_n | \vec{U}_{n-1}, C \neq C(U_n)) = 0$

We have

- $P(U_n | \vec{U}_{n-1}) = P(U_n | C(U_n), \vec{U}_{n-1}) * P(C(U_n) | C(U_1), C(U_2), \dots, C(U_{n-1}))$

Quantized Dialog Algorithm

Problem Statement

Given $U_{trn}(i, 1)$, $U_{trn}(i, 2)$, ... , $U_{trn}(i, n_i)$ training utterances for conversations $i = 1, \dots, I$, build a conversational system that can accurately predict the next utterance $U_{tst}(2j)$ having observed the conversation context $U_{tst}(1)$, $U_{tst}(2)$, ... , $U_{tst}(2j - 1)$

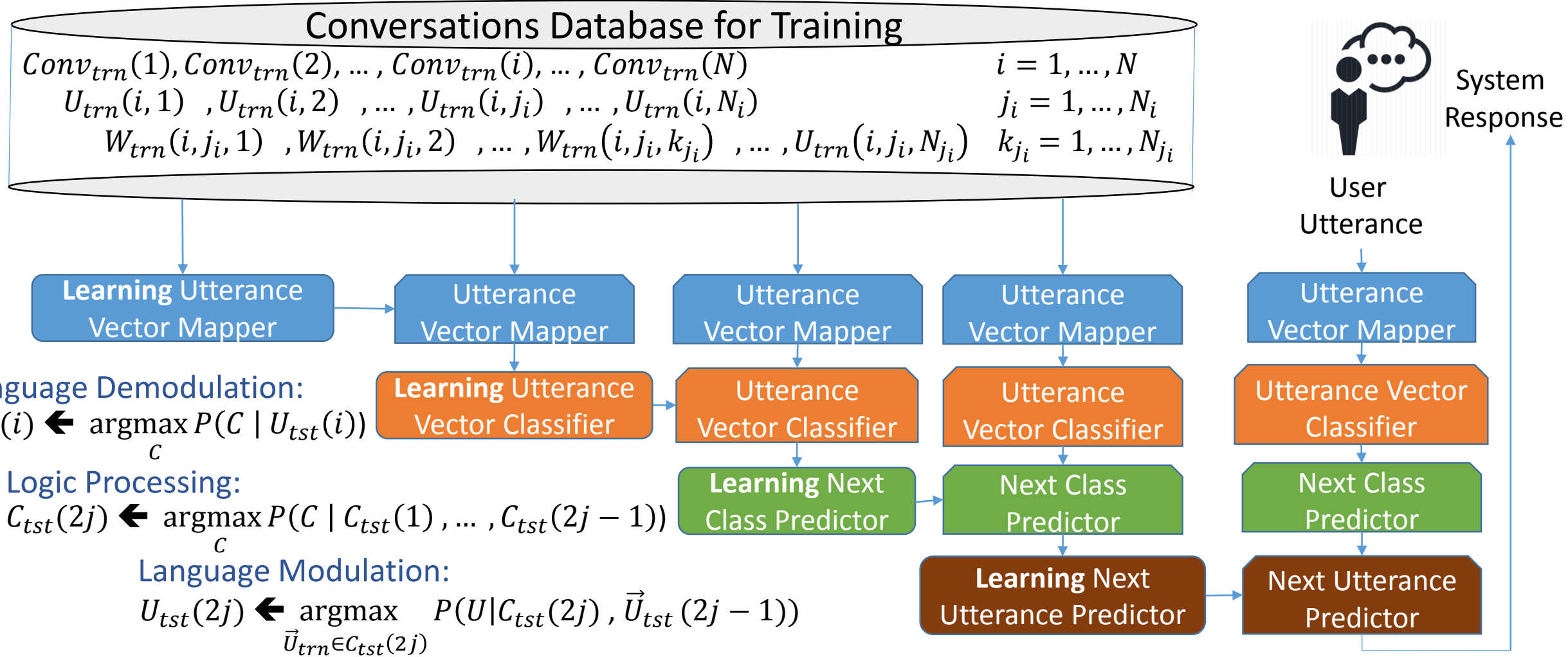
Training

- **Step 1:** Cluster the utterance pool into the smallest number of clusters. Reinforce the notion of Language Demodulation
- **Step 2:** Replace the utterance sequences $U_{trn}(i, 1)$, ... , $U_{trn}(i, n_i)$ with their cluster sequence $C_{trn}(i, 1)$, ... , $C_{trn}(i, n_i)$
- **Step 3:** Build your favorite model for optimally estimating $P(C_{trn}(i, j) \mid C_{trn}(i, 1), C_{trn}(i, 2), \dots, C_{trn}(i, j - 1))$
- **Step 4:** Build your favorite Seq2Seq system for optimally calculating $P(U_{trn}(i, 2j) \mid C_{trn}(i, 2j), \vec{U}_{trn}(i, 2j - 1))$

Testing

- **Step1:** Label utterances of conversation $U_{tst}(1), \dots, U_{tst}(2j - 1)$ to obtain the cluster sequence $C_{tst}(1), \dots, C_{tst}(2j - 1)$
- **Step2:** Find $C_{tst}(2j) \leftarrow \arg\max_C P(C \mid C_{tst}(1), C_{tst}(2), \dots, C_{tst}(2j - 1))$
- **Step 3:** Find $U_{tst}(2j) \leftarrow \arg\max_{\vec{U}_{trn} \in C_{tst}(2j)} P(U \mid C_{tst}(2j), \vec{U}_{tst}(2j - 1))$

Quantized Dialog Training & Runtime Processes



Some Early Results

DSTC6 (US Team)
Ubuntu (India Team)

DSTC6 – Goal Oriented Dialog Track

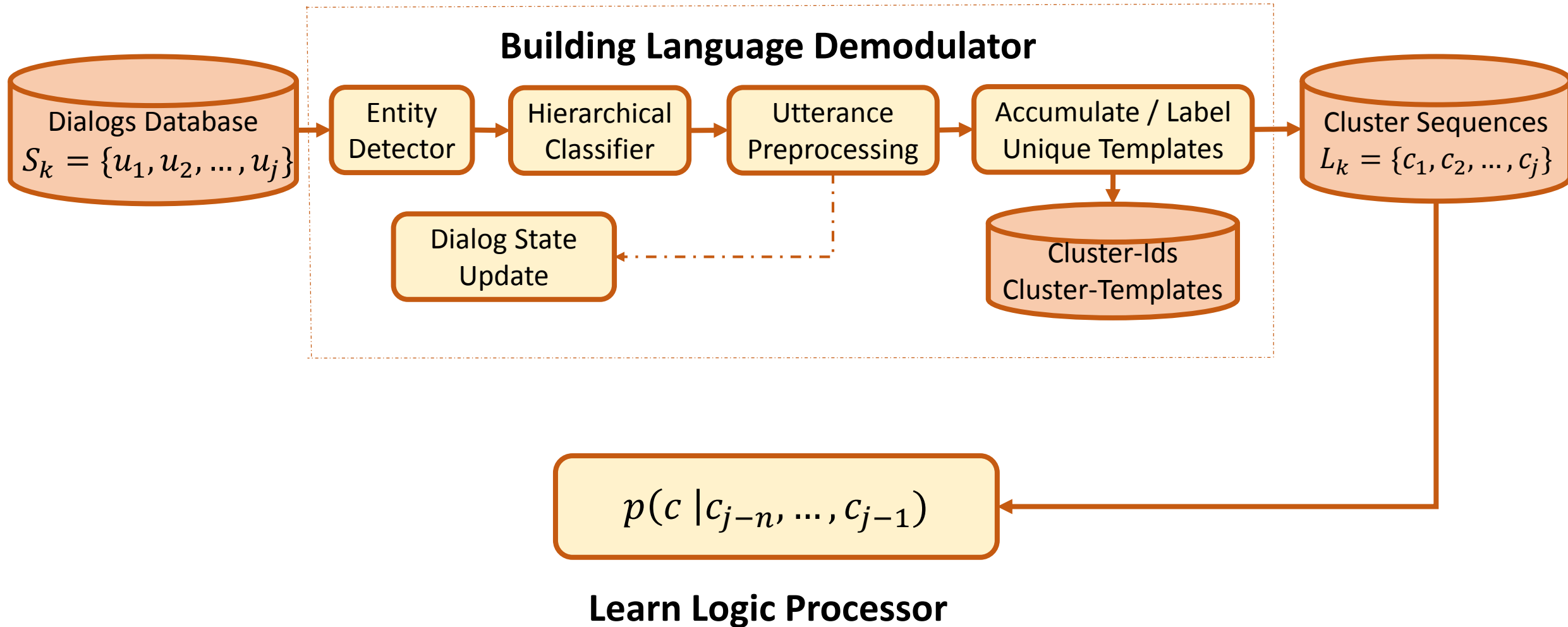
- Restaurant-reservation problem
 - The objective is to book a table in a restaurant satisfying a number of requirements given by the user
- Challenge is divided into 5 Tasks
 1. Issuing API calls – 5 slots: “cuisine”, “location”, “price range”, “number of people in the party”, “atmosphere”
 2. Updating API calls – addressing user’s change of mind
 3. Displaying options – helping user select a restaurant among possible options
 4. Providing extra information - address and/or phone number of the selected restaurant
 5. Conducting full dialog – Combine Tasks 1-4 to generate full dialogs
- Data
 - Train: 50,000 dialogs split equally (10,000 dialogs) over the 5 Tasks
 - Test: 4 test sets, each with 1,000 dialogs split for each of 5 Tasks
 - Test set 1 – No change in knowledgebase
 - Test set 2 – Knowledgebase has different values for the same 5 slots
 - Test set 3 – Knowledgebase has an added slot in the API calls: “Restrictions”
 - Test set 4 – Knowledgebase has the added “Restrictions” slot and different values for all slots

Quantized Dialog for DSTC6 Task

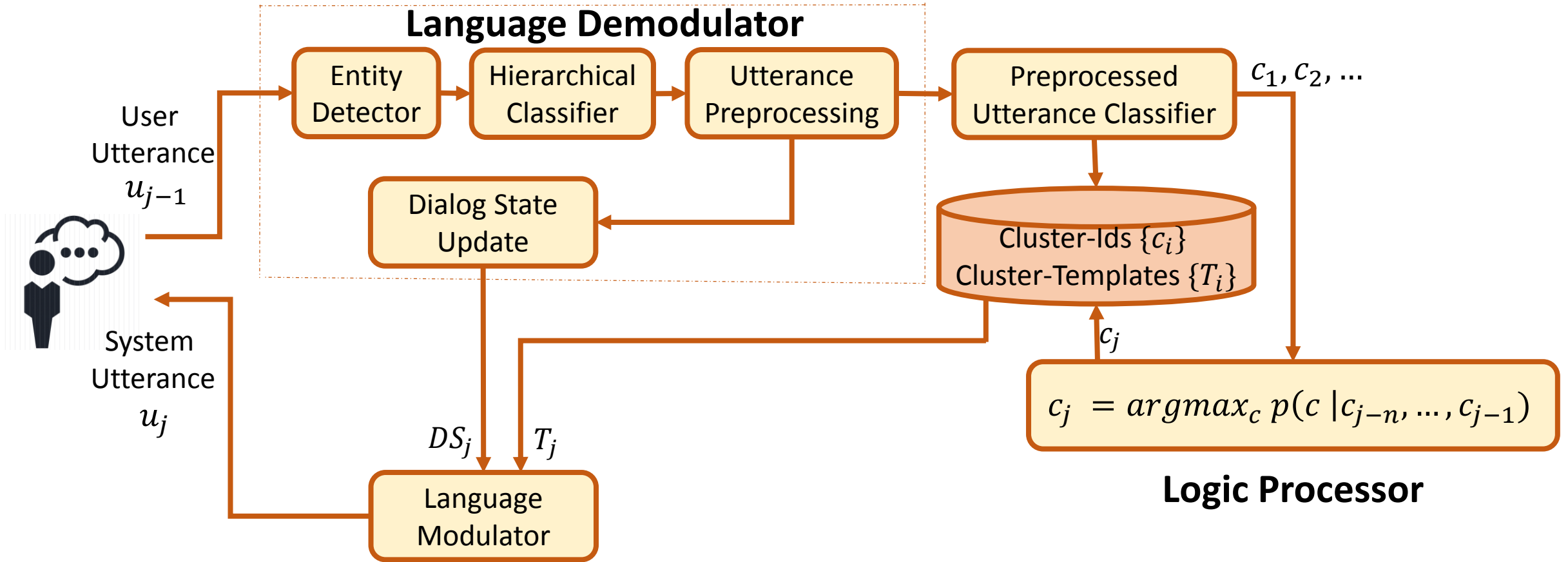
Contract the Language to the highest possible Logic common denominator

- **Delexicalized** –
“I like *Mexican*”, “I like *Persian*” ➔ “I like *Food_Type*”
- **Paraphrase** –
“I *prefer* Mexican”, “I *like* Mexican” ➔ “I *Want* Mexican”
- **Order** –
“Should we eat at *Daniel* or *Per Se*? ➔ “Should we eat at *Rest_1* or *Rest_2*?”
- **AV Pair** –
“No. of people in your *group*?”, “What *kind* of *food*?” ➔ “What is the *Value* of the *Type*?”

DSTC6 Task Quantized Dialog Training Process



DSTC6 Task Quantized Dialog Runtime Process



$u_j =$ deterministic modulation of entity values on T_j

DSTC6 Quantized Dialog Hierarchical Classifier

Classifier 1:

Identifies the utterances with 'not confirmed' entities

Example:

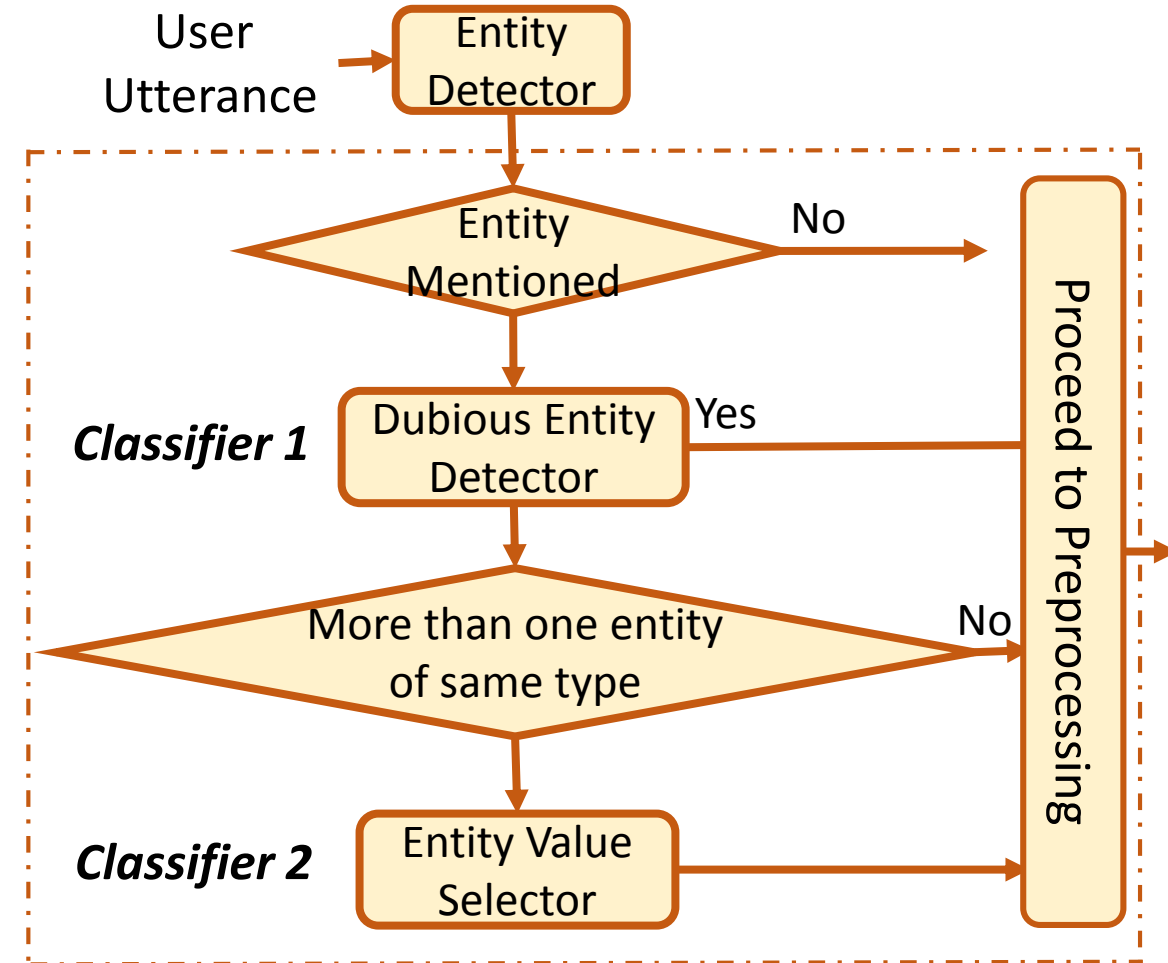
- I. *Expensive* is tempting but *moderate* may be more reasonable – *dubious*
- II. *Expensive* is very tempting but I'm afraid I have to stick to *moderate* today – *confirming*

Classifier 2:

Identifies the correct entity, when more than one entity is specified

Example:

- I. let's go with *Chinese* food, I will keep *Indian* for next time (original) →
- II. let's go with *ENTITY_0* food, I will keep *ENTITY_1* for next time (preprocessed) – *ENTITY_0* (Chinese)



DSTC6 Quantized Dialog – Preprocessing and DS update

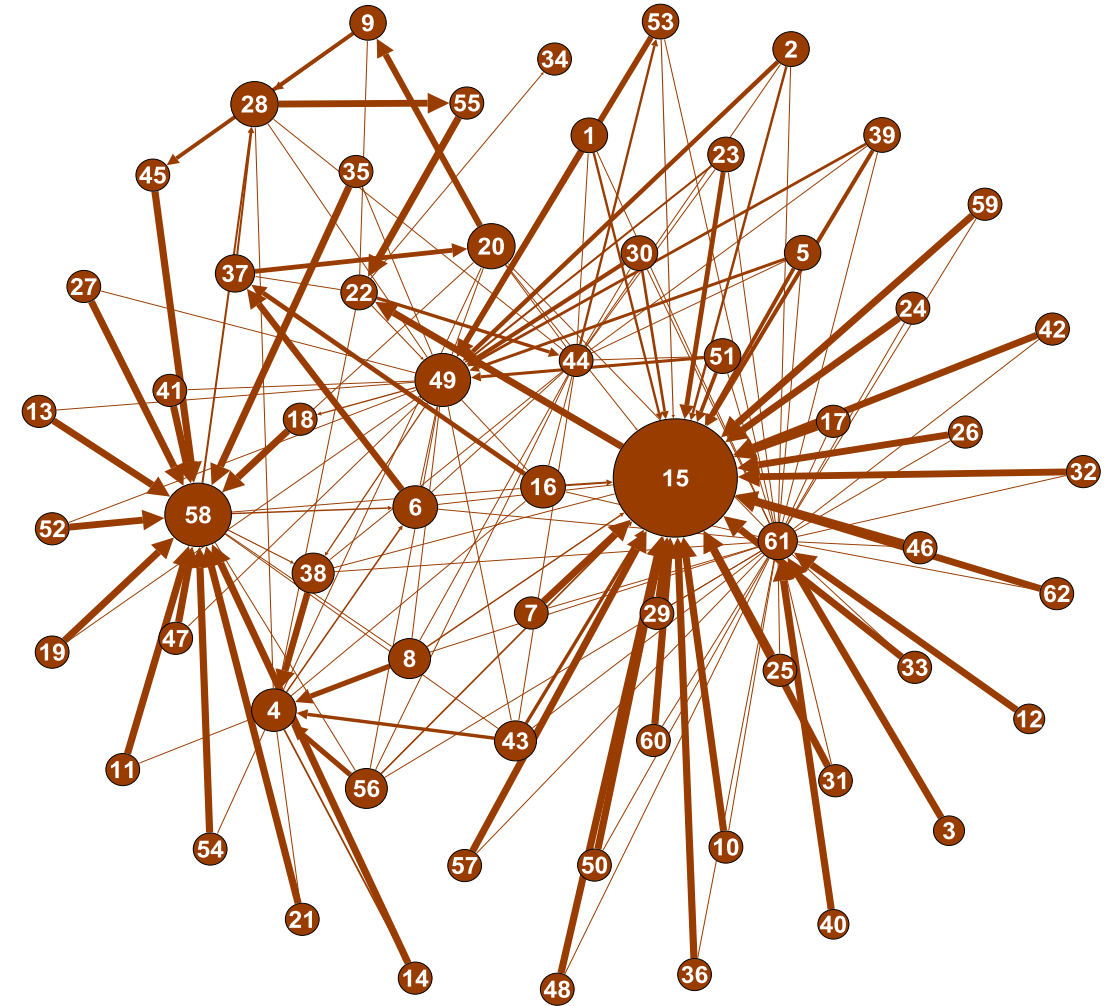
- User utterances with entity values transformed into corresponding vector of entity types
- The dialog-state update is based on a keyword matching algorithm, assisted by a hierarchical sentence classifier

Turn	Original Utterance	Preprocessed representation	Dialog state
1	hello	hello	{}
2	hello what can i help you with today	hello what can i help you with today	{}
3	i'd like to book a table with a business atmosphere with spanish cuisine	CUISINE_TYPE NONE NONE NONE ATMOSPHERE	CUISINE_TYPE : spanish, ATMOSPHERE : business
4	i am on it	i am on it	CUISINE_TYPE : spanish, ATMOSPHERE : business
5	<silence>	silence	CUISINE_TYPE : spanish, ATMOSPHERE : business
6	where should it be	where should it be	CUISINE_TYPE : spanish, ATMOSPHERE : business
7	find me one in bombay, paris will be too complicated	CUISINE_TYPE LOCATION NONE NONE ATMOSPHERE	CUISINE_TYPE : spanish, ATMOSPHERE : business, LOCATION : bombay

Visualization of Clusters and Transitions

Task	Number of Unique Sentences	Number of clusters after first preprocessing
1	6267	62
2	11166	15
3	11753	57
4	8409	41
5	13848	156

- Each node corresponds to a cluster in task 1
- The size of a node is proportional to the no. of incoming edges
- The width of an edge is proportional to the transition probability



DSTC6 Quantized Dialog n-gram Model

- Consider dialog $D = \{u_1, u_2, \dots, u_j\}$ where u_i is the i_{th} utterance
- In the quantized space, $D = \{c_1, c_2, \dots, c_j\}$ where c_i is the clusterId for utterance u_i
- ClusterId sequences for all training dialogs are used to construct an n-gram Model for $p(c | c_{j-n}, \dots, c_{j-1})$
- Utterance prediction problem of $u_j = \operatorname{argmax}_v p(v | u_1, \dots, u_{j-1})$ is formalized in 2 steps:
 1. $c_j = \operatorname{argmax}_c p(c | c_{j-n}, \dots, c_{j-1})$
 2. $u_j =$ deterministic modulation of entity values on T_j (linguistic template of c_j)

DSTC6 with other Methods

- We built Non-Quantized Dialog systems using MemN2N and FFNN
- We used 2500 dialogs for each task as held out data for evaluation
- The results of these methods for the 2500 held out data were not as good as QDNGRM
- We decided to run Quantized Dialog with MemN2N, FFNN, CNN, & LSTM for learning $p(c | c_{j-n}, \dots, c_{j-1})$
- Preprocessing and post processing remained the same
- As expected, we got similar results as in QDNGRM

Quantized Dialog MemN2N Model

- Input represented as bag-of-clusterIds of “all” the past utterances in a dialog, excluding last one
 $c_1, c_2, \dots, c_{j-1}$
- Last utterance clusterId was used as the “Question”
- As in QDNGRM, output was over the entire clusterId set
- Matrices A & C are learned. B is tied to A
- Learn two embedding vectors for each clusterId
- Embedding vectors size were set at 30
- A 3 layer MemN2N was constructed
- Trained on 75% of training data and evaluated on the 25%

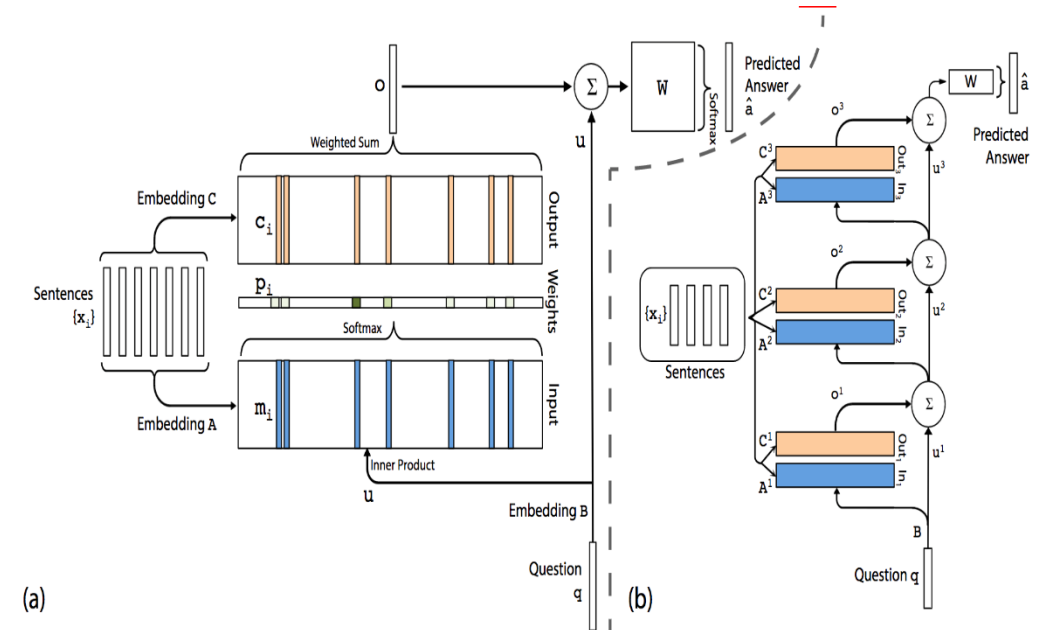
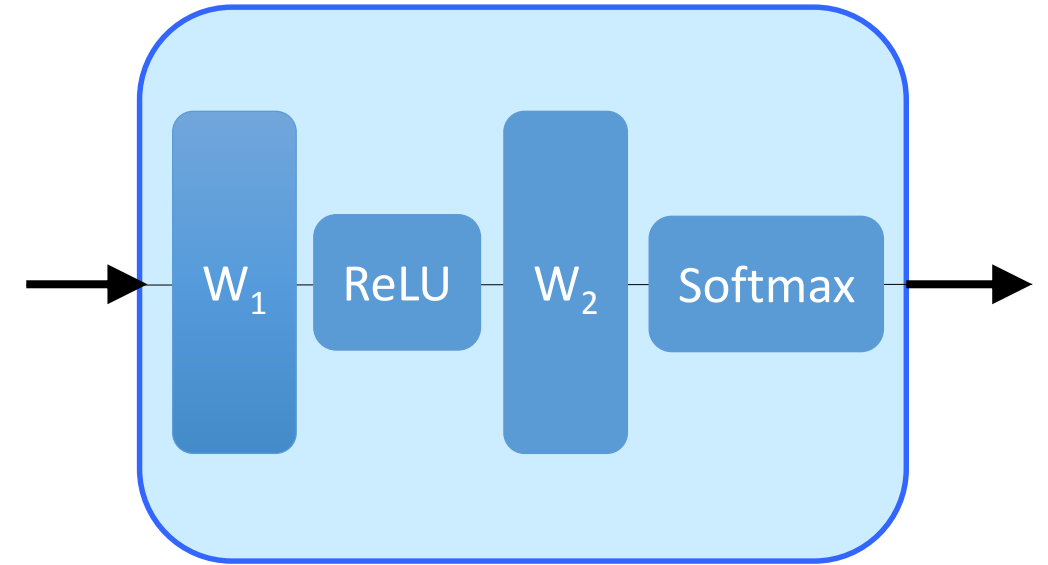


Figure 1: (a): A single layer version of our model. (b): A three layer version of our model. In practice, we can constrain several of the embedding matrices to be the same (see Section 2.2).

Image from <https://papers.nips.cc/paper/5846-end-to-end-memory-networks.pdf>

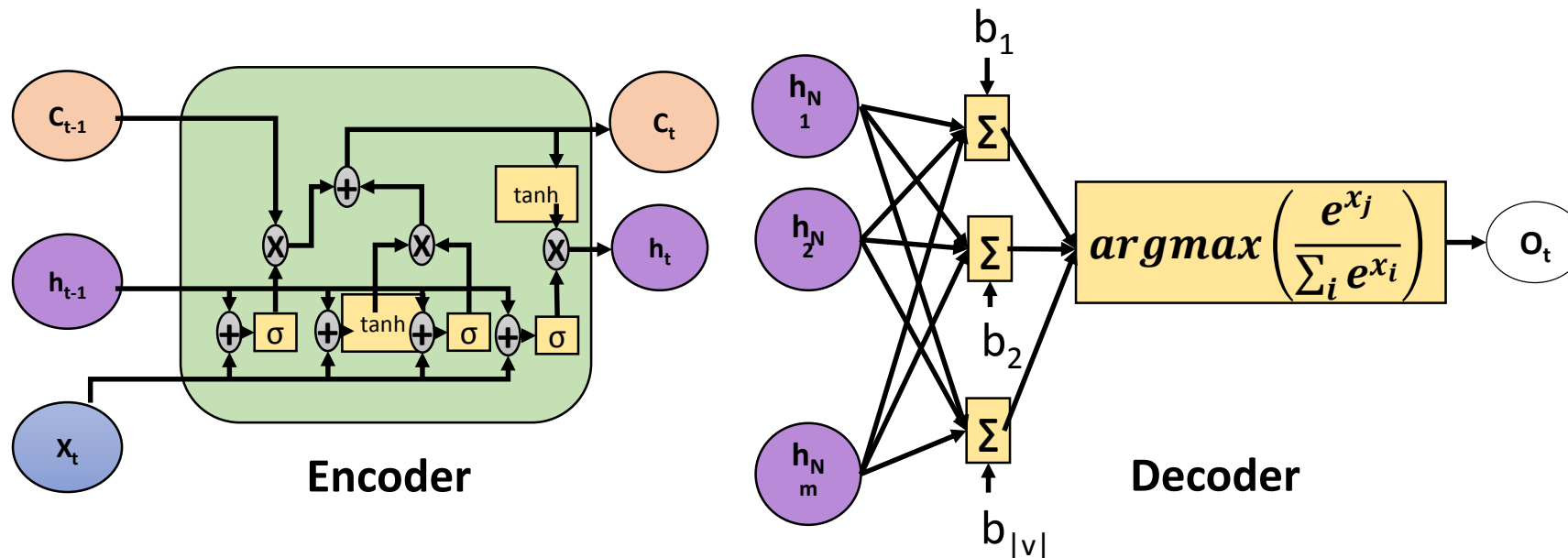
Quantized Dialog FFNN Model

- Up to the last 6 utterances were used
- 100-dim embedding vectors were learned for each ClusterId
- 6 of these vectors concatenated as input to W_1
- W_1 : 600 x 100
- W_2 : 100 x Size of clusterId set
- Neural network trained on 75% of dataset and tested on remaining 25%
- Cross-entropy; Adam optimizer; Convergence after ~200 epochs



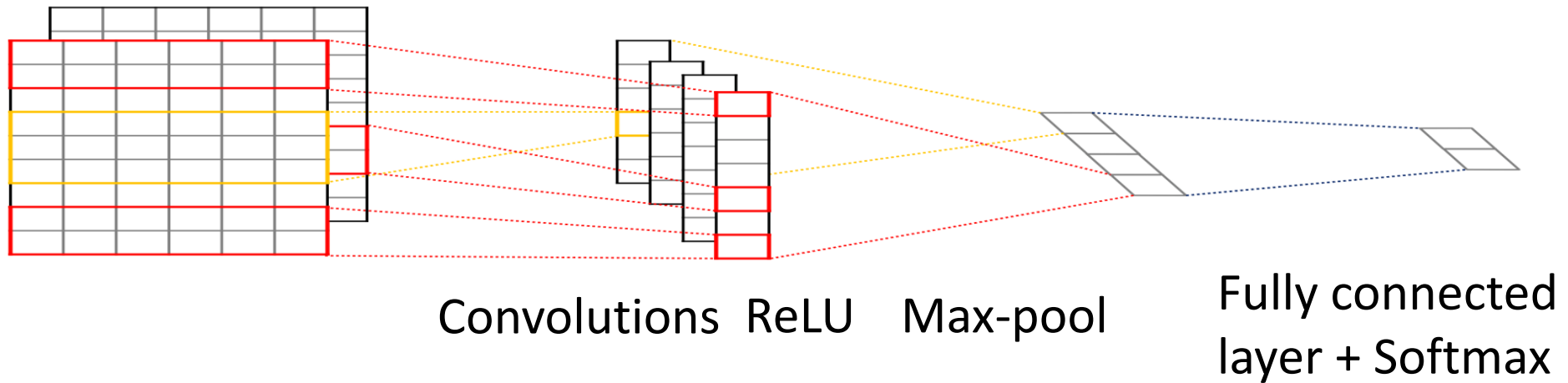
Quantized Dialog LSTM Model

- Encoder-Decoder architecture
- LSTM encoder*
- Linear + Softmax decoder
- 100-dim vector embedding learned for each clusterId
- Neural network trained on 75% of dataset and tested on remaining 25%
- Cross-entropy; Adam optimizer; Convergence after ~500 epochs



Quantized Dialog CNN Model

- 30-dim embedding vectors learned for each clusterId
- Convolutions were performed with 16 different filters of size 5x30
- Entire ClusterId sequence was used as the history
- Network trained on 75% of data and evaluated on the remaining 25%



Quantized Dialog Results

- The dataset corresponding to each task was divided into two components : 75% - training data, 25% - test data
- Following numbers show prediction accuracy (%) for each task

Task	Number of clusters	N-gram Language model	FFNN		LSTM		MemN2N		CNN	
			Training	Testing	Training	Testing	Training	Testing	Training	Testing
1	62	99.9	100	100	100	100	100	100	100	99.8
2	15	100	100	100	100	100	100	100	100	100
3	57	100	100	99.9	99.9	99.7	100	100	100	99.9
4	41	100	100	100	100	100	100	100	100	99.9
5	156	100	97.9	97.8	99.5	99.1	100	99.9	100	99.8

QDNGRM Results on DSTC6 Test Data

With initial quantization (submitted)

Task	tst1	tst2	tst3	tst4
1	100	100	86.6	88.2
2	100	100	100	100
3	99.8	99.7	100	100
4	100	100	100	100
5	98.6	98.5	95.7	95.3

With further quantization

Task	tst1	tst2	tst3	tst4
1	100	100	99.6	99.9
2	100	100	100	100
3	100	99.9	100	100
4	100	100	100	100
5	99.6	99.6	99.9	99.9

Task	Number of Unique Sentences	Number of clusters after first preprocessing	Number of clusters after second preprocessing
1	6267	62	58
2	11166	15	14
3	11753	57	39
4	8409	41	18
5	13848	156	98

Quantized Dialog as a Dialog System

- QDNGRM trained on 75% conversations in task 5 data was used to predict the system utterance for the other 25% at each turn using a 7-gram language model

System turn	Number of failures
1-5	0
6-10	0
11-15	3
16-20	6
21-25	5
25-30	1
Last turn	0

- QDNGRM has an overall accuracy of 99.4% when used as a dialog system

Some Early Results

DSTC6 (US Team)
Ubuntu (India Team)

Ubuntu Corpus V2¹

Data format: (Context, Response, Flag)

Context	Response	Flag
well, can I move the drives? __eot__ ah not like that	I guess I could just get an enclosure and copy via USB	1
well, can I move the drives? __eot__ ah not like that	you can use “ps ax” and “kill (PID #)”	0

Training Data: 1M triples (500K positive (flag=1), 500K negative (flag=0))

Validation Data: 19,560 points with 1 GT and 9 incorrect agent utterances

Test Data: 18,920 points with 1 GT and 9 incorrect agent utterances

TASK: Select one out of 10 possibilities

¹<https://github.com/rkadlec/ubuntu-ranking-dataset-creator> , Lowe et al., 2017

Retrieval based End2End Dialog Systems

- Do not have the luxury of working with only 10 candidates
- We need more realistic systems where number of candidates is *large*
- Increased the number of candidates to 1000
 - One candidate is Ground Truth and 999 are selected randomly from training data
- Besides accuracy, we also have computation challenge when N is **very large**
- Initial results on pruning the candidates

Quantized Dialog End2End Retrieval System

- Restrict the number of viable candidates by using a concept based Quantization
- **Step 1:** Given a conversation context *predict* plausible concepts for agent utterance
 - Concepts are learnt by training an LDA topic model on training conversations
- **Step 2:** Filter candidates by restricting them to predicted concepts.
- **Step 3:** Use dual encoder to rank the remaining candidates.

How much reduction can we achieve while preserving accuracy?

Results so far ...

Model	R@1	R@2	R@5	MRR ¹	Compression ²
Dual Encoder	0.096	0.148	0.234	0.171	1.0
Topic Selection + Dual Encoder 1000 Topics, Select Top 50 topics	0.097	0.148	0.236	0.171	0.7448

- Results with 1000 candidates (of which 1 is correct)
- ***After topic selection***, we obtain similar fidelity and only ***process about 75%*** of the candidates.

¹MRR: Mean Reciprocal Rank (Computed using 1000 candidates)

²Compression: #Candidates processed / Total Candidates

Summary

- Far from self-organizing design that learns thru guidance, observation, & experimentation
- Model each utterance in a conversation as a carrier-language modulated by an embedded-logic
- Contract the Language to the highest possible Logic Common Denominator
 - Demodulate the carrier-language in an utterance by Quantizing the utterance
 - Learn a Statistical Logic Processor in the quantized space
- Finding the appropriate language free quantized space is not easy
- Initial success of Quantized Dialog for simple DSTC6 Goal Oriented Task and preliminary results of Quantized Retrieval on Ubuntu task are promising