Globally Coherent Text Generation with Neural Checklist Models

Chloe´ Kiddon, Luke Zettlemoyer, Yejin Choi

Computer Science & Engineering
University of Washington

Presenter: Webber Lee

March 29, 2018

Outline

- Introduction
- Previous work
- Task description
- Proposed model
- Experimental results
- Conclusion

Introduction

- Recurrent neural network (RNN) has been proven to be well suited for many natural language generation tasks
- Problems:
  - Can miss information
  - Can introduce duplicated or superfluous content
  - Common when
    - There are multiple distinct sources of input
    - Length of output text is long
- Example: generating a cooking recipe
  - Input: title and ingredient list
  - Output: complete text that describes how to produce desired dish
  - Problem: may lose track of which ingredients have already been mentioned

Previous work

- Attention models have been used for many NLP tasks
  - used to record what has been said and to select new agenda items
- Previous works focus on generating short texts and assume fixed set of agenda items
  - Composes longer texts with a more varied and open ended set of agenda items
- Other challenges:
  - Maintain coherence
  - Avoid duplication
  - …

Task description

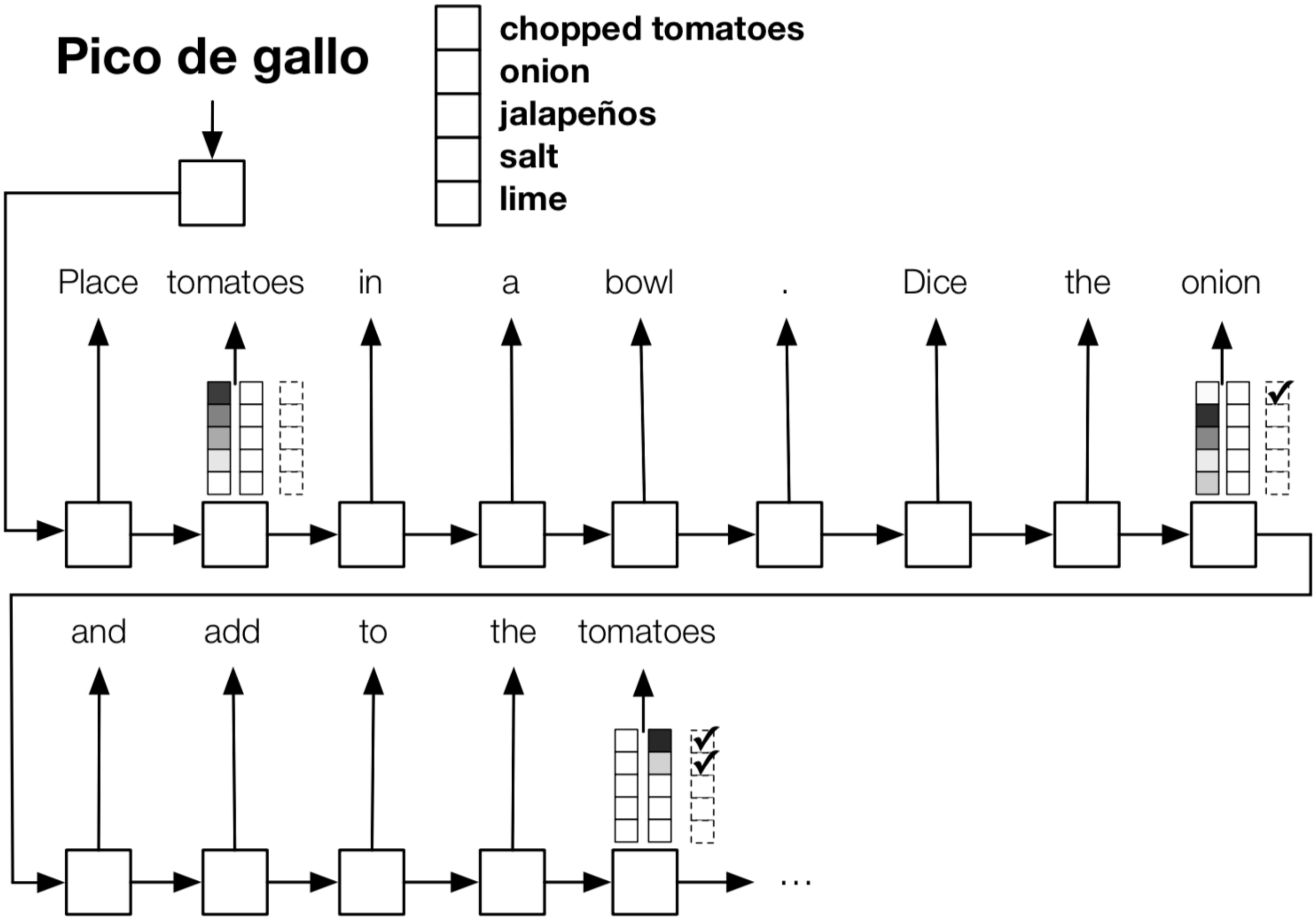
- Input:
  - A goal *g*
    - ex1: Recipe generation; recipe title; “pico de gallo”
    - ex2: Dialogue system; dialogue type; “inform” or “query”
  - An agenda $E = \{e_1, e_2, \ldots, e_{|E|}\}$
    - ex1: ingredient list; “lime,” “salt”
    - ex2: hotel name, address, or details
- Output:
  - A goal-oriented text *x*
    - ex1: Mix the turkey with flour, salt…
    - ex2: Hotel Stratford does not have internet

Neural checklist model

- Goal: generate a recipe for a particular dish while keeping track of an agenda of items (list of gradients) to be mentioned

- The model learns interpolate among three components at each time step:
  - An encoder-decoder language model to generate goal-oriented texts
  - An attention model that tracks remaining agenda items to be introduced
  - An attention model that tracks used or checked agenda items

Example checklist recipe generation



Definitions of proposed model

Given

- Goal embedding: $\mathbf{g} \in \mathbb{R}^k$
- Matrix of *L* agenda items: $E \in \mathbb{R}^{L \times k}$
- Checklist of what items have been used: $\mathbf{a}_{t-1} \in \mathbb{R}^L$
- Previous hidden state: $\mathbf{h}_{t-1} \in \mathbb{R}^k$
- Current input word embedding: $\mathbf{x}_t \in \mathbb{R}^k$

Computes

- Next hidden state: $\mathbf{h}_t$
- Embedding used to generate output word: $\mathbf{o}_t$
- Updated checklist: $\mathbf{a}_t$

Diagram of neural checklist model

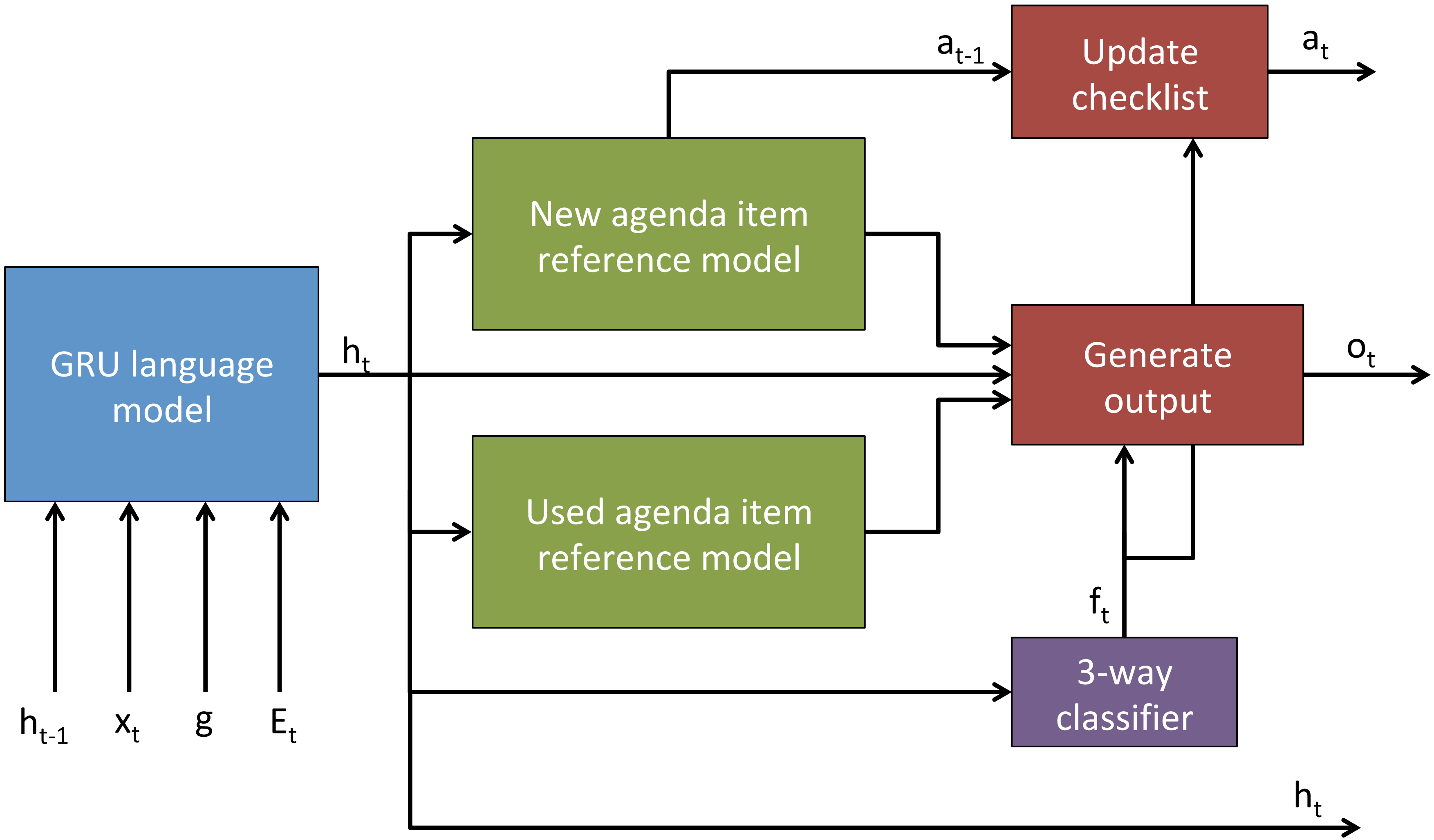
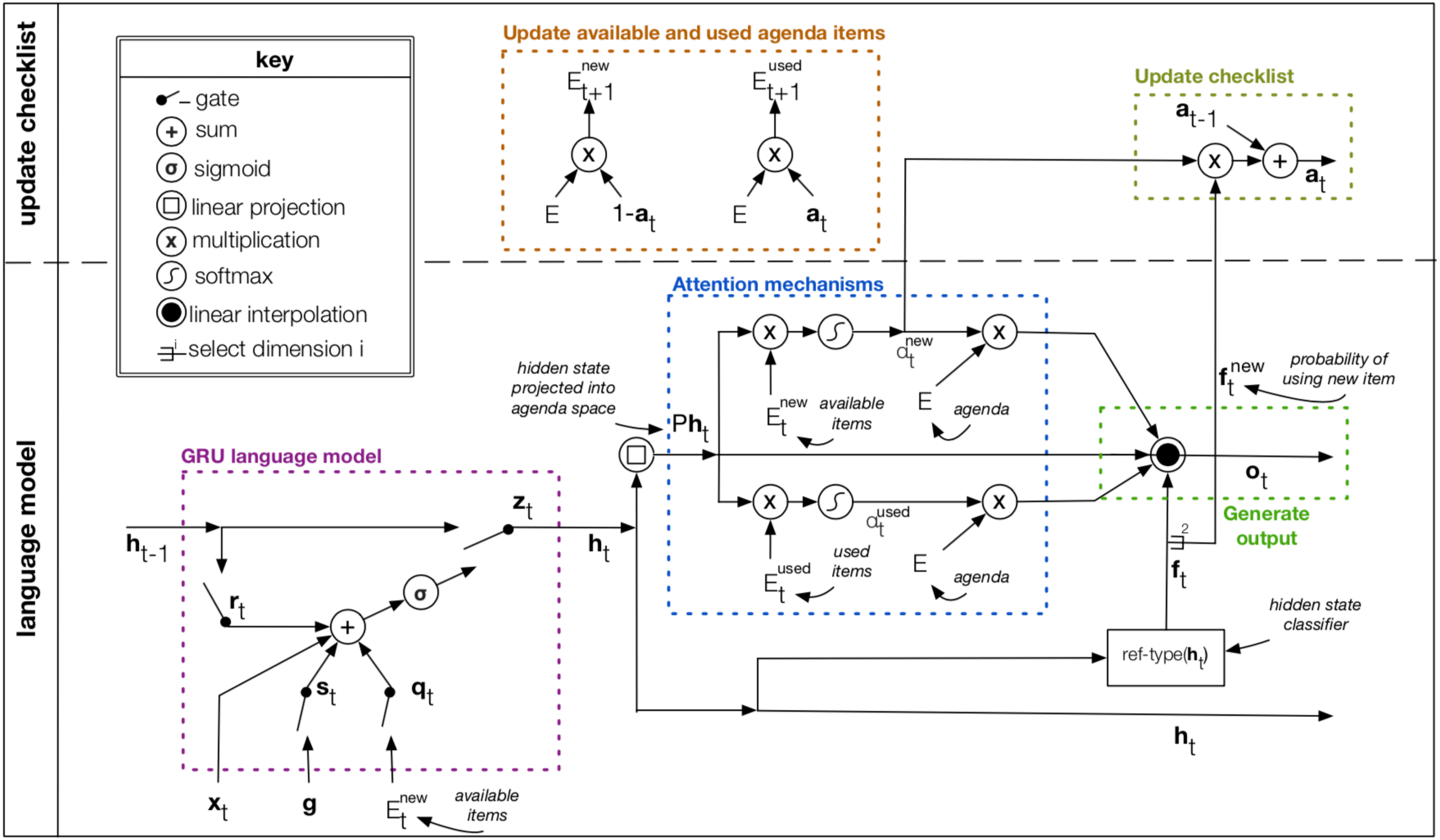


Diagram of neural checklist model

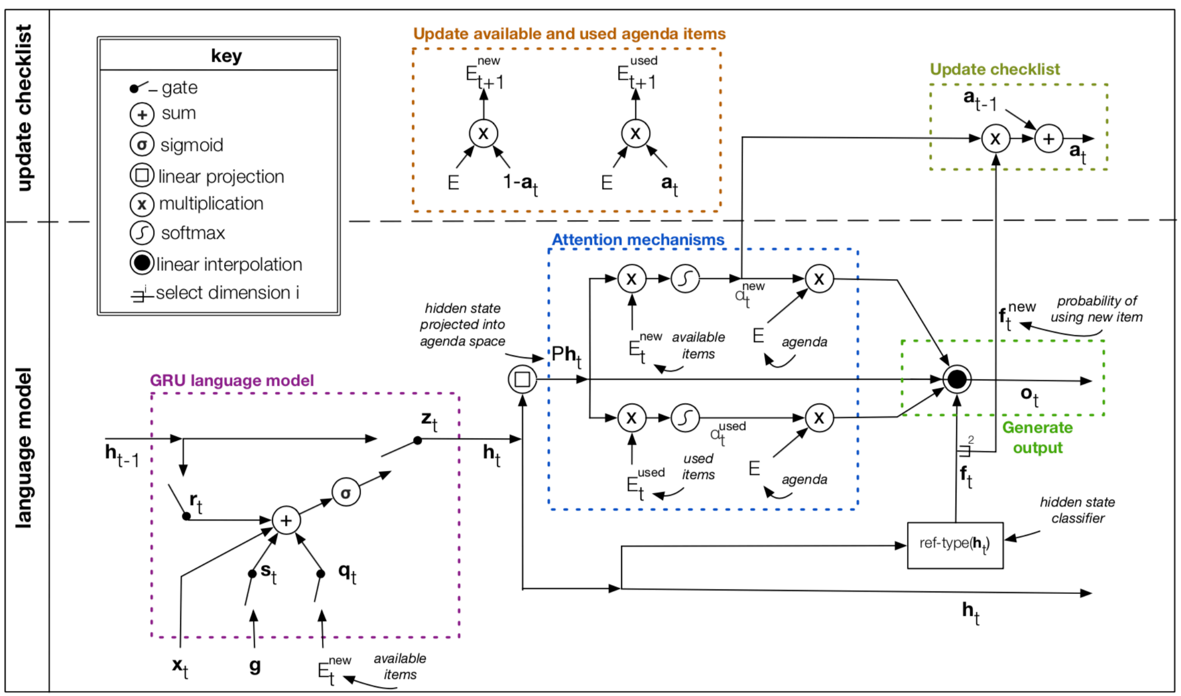


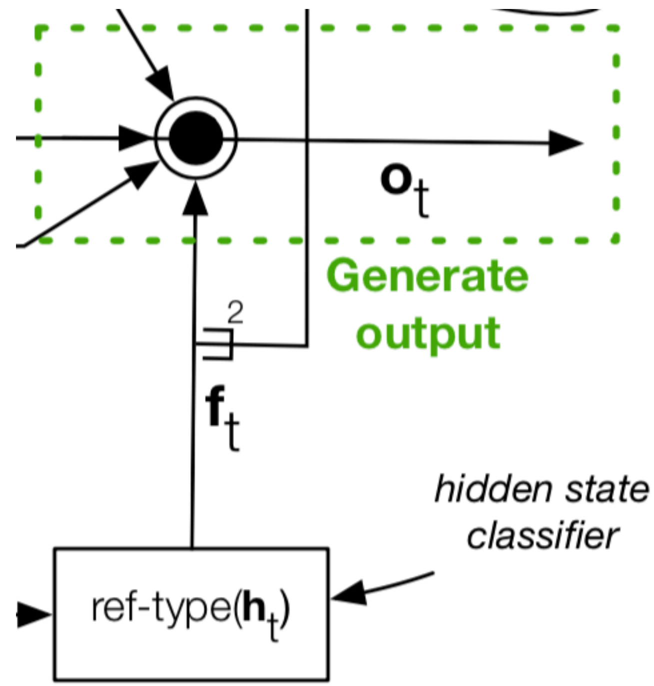
Generating output token probabilities

- Project output hidden state O_t into vocabulary space

$$\mathbf{w}_t = \mathrm{softmax}(W_o \mathbf{o}_t)$$

  - W_o is a trained projection matrix





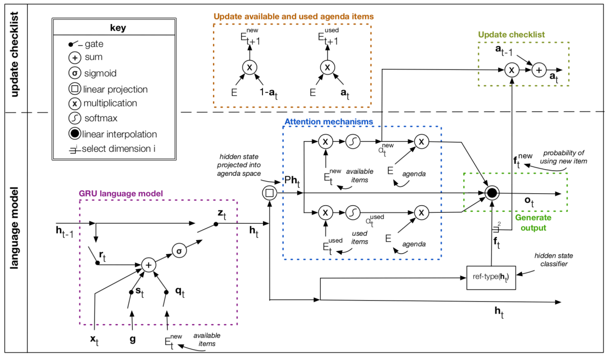
Generating output token probabilities

- Output hidden state is the linear interpolation of
  - c_t^{gru}: content from Gated Recurrent Unit (GRU)
  - c_t^{new}: encoding from new agenda item reference model
  - c_t^{used}: encoding from previously used item model

$$\mathbf{o}_t = f_t^{gru}\mathbf{c}_t^{gru} + f_t^{new}\mathbf{c}_t^{new} + f_t^{used}\mathbf{c}_t^{used}$$

  - $f_t = [f_t^{gru}, f_t^{new}, f_t^{used}]$ is interpolation weights learned by a three-way probabilistic classifier

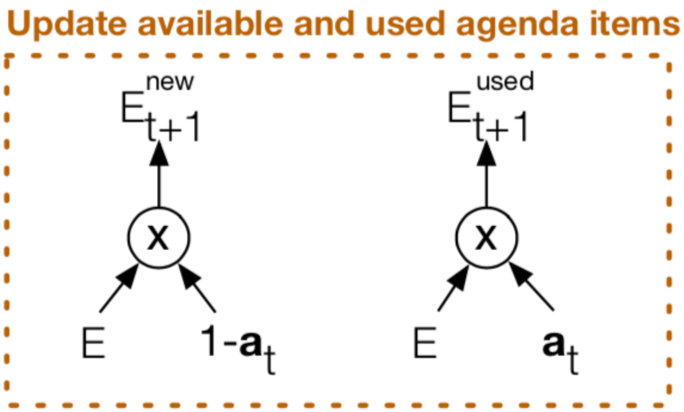
$$\mathbf{f}_t = ref\text{-}type(\mathbf{h}_t) = softmax(\beta S\mathbf{h}_t)$$



New and used agenda item reference models

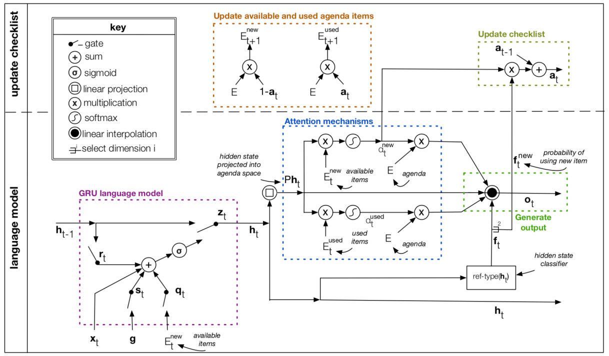
- Key features:
  - predicts which agenda item is being referred to
  - stores those predictions for use during generation
- Checklist vector a_t represents the probability each agenda item has been introduced into the text
  - initialized to all zero at t = 1
- Renaming/used item matrices

$$E_t^{new} = ((\mathbf{1}_L - \mathbf{a}_{t-1}) \otimes \mathbf{1}_k) \circ E$$
$$E_t^{used} = (\mathbf{a}_{t-1} \otimes \mathbf{1}_k) \circ E$$



  - $\otimes$ replicate L-dimensional vector by k times (i.e., $R^L \rightarrow R^{L \times k}$)
  - $\circ$ element-wise multiplication

Agenda item reference models (cont)



- The alignment is probability distribution representing how close h_t is to each item

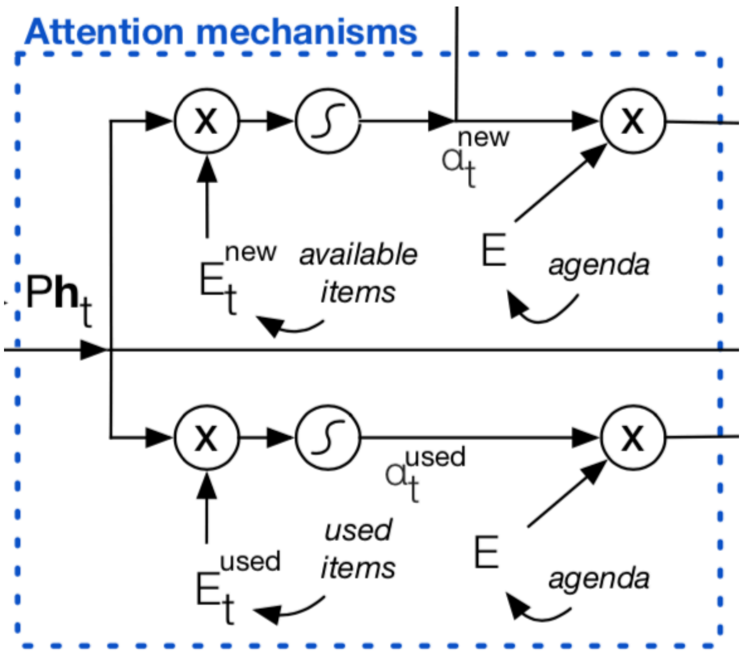
$$\boldsymbol{\alpha}_t^{new} \propto \exp(\gamma E_t^{new} P \mathbf{h}_t)$$

$$\boldsymbol{\alpha}_t^{used} \propto \exp(\gamma E_t^{used} P \mathbf{h}_t)$$

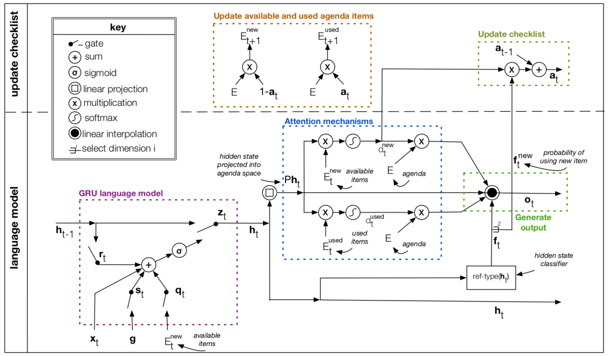
- The attention encoding is the attention-weighted sum of agenda items

$$\mathbf{c}_t^{new} = E^T \boldsymbol{\alpha}_t^{new}$$

$$\mathbf{c}_t^{used} = E^T \boldsymbol{\alpha}_t^{used}$$

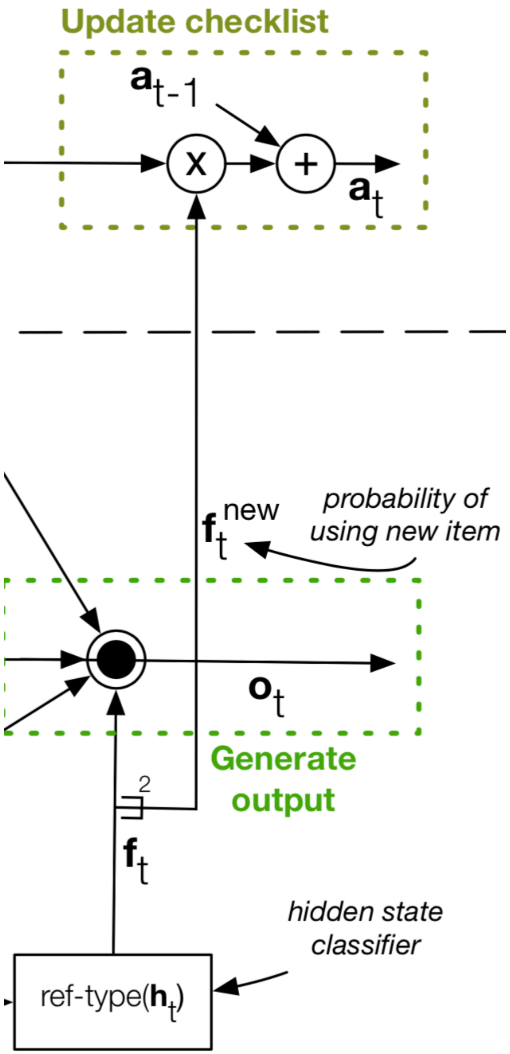


Agenda item reference models (cont)



- Checklist update

$$\mathbf{a}_t^{new} = \mathbf{f}_t^{new} \cdot \boldsymbol{\alpha}_t^{new}$$
$$\mathbf{a}_t = \mathbf{a}_{t-1} + \mathbf{a}_t^{new}$$



Review of GRU model

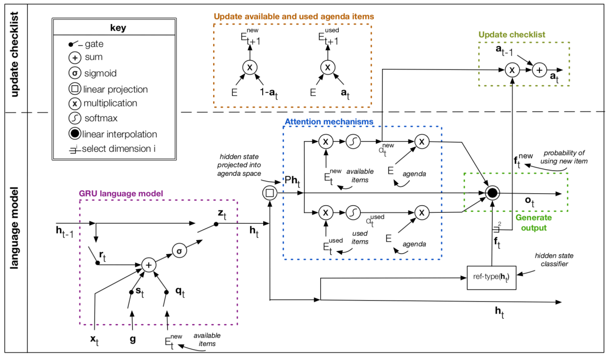
$$\mathbf{r}_t = \sigma(W_r \mathbf{x}_t + U_r \mathbf{h}_{t-1})$$

$$\mathbf{z}_t = \sigma(W_z \mathbf{x}_t + U_z \mathbf{h}_{t-1})$$

$$\tilde{\mathbf{h}}_t = \tanh(W \mathbf{x}_t + \mathbf{r}_t \odot U \mathbf{h}_{t-1})$$

$$\mathbf{h}_t = (\mathbf{1} - \mathbf{z}_t)\mathbf{h}_{t-1} + \mathbf{z}_t \tilde{\mathbf{h}}_t$$

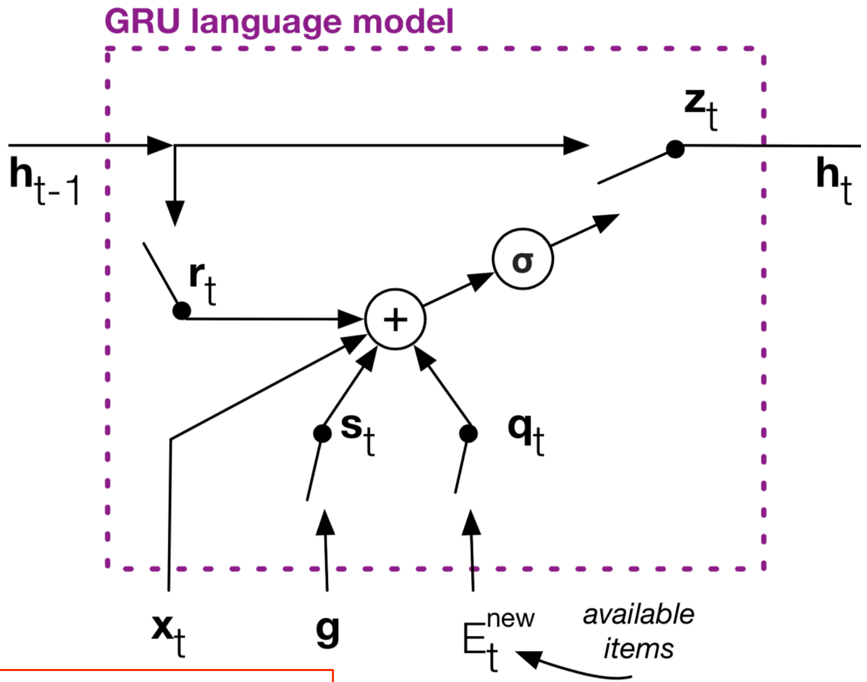
Modified GRU model



$$\mathbf{r}_t = \sigma(W_r \mathbf{x}_t + U_r \mathbf{h}_{t-1})$$

$$\mathbf{z}_t = \sigma(W_z \mathbf{x}_t + U_z \mathbf{h}_{t-1})$$

$$\boxed{\begin{aligned}\mathbf{s}_t &= \sigma(W_s \mathbf{x}_t + U_s \mathbf{h}_{t-1}) \\ \mathbf{q}_t &= \sigma(W_q \mathbf{x}_t + U_q \mathbf{h}_{t-1})\end{aligned}}$$



$$\boxed{\begin{aligned}\tilde{\mathbf{h}}_t \quad = \quad & \tanh(W_h x_t + \mathbf{r}_t \odot U_h \mathbf{h}_{t-1} \\ & + \mathbf{s}_t \odot Y\mathbf{g} + \mathbf{q}_t \odot (\mathbf{1}_L^T Z E_t^{new})^T\end{aligned}}$$

$$\mathbf{h}_t = (\mathbf{1} - \mathbf{z}_t)\mathbf{h}_{t-1} + \mathbf{z}_t \tilde{\mathbf{h}}_t$$

Experimental Setup

- Implemented and trained using Torch framework
- Two tasks: (1) recipe generation (2) dialogue responses
- Parameters
  - gradient norm: 0.5; uniformly on [-0.35, 0.35]
  - beam search size: 10
  - learning rate: 0.1
  - temperature hyper-parameters (beta, gamma)
    - recipe: (5,2)
    - dialogue: (1, 10)
  - hidden state size
    - recipe: 256; dialogue: 80
  - batch size
    - recipe 30; dialogue: 10

Quantitative results on recipe task

- You're Cooking recipe library
  - 82,590 recipes used for training; 1000 for development and testing
- BLEU and METEOR are not good metrics for this task

Model	**BLEU-4**	**METEOR**	**Avg. % given items**	**Avg. extra items**
Attention	2.8	8.6	22.8%	3.0
EncDec	3.1	9.4	26.9%	2.0
NN	**7.1**	12.1	40.0%	4.2
NN-Swap	**7.1**	**12.8**	58.2%	2.1
Checklist	3.0	10.3	67.9%	**0.6**
- $\mathbf{o}_t = \mathbf{h}_t$	2.1	8.3	29.1%	2.4
- no used	3.0	10.4	62.2%	1.9
- no supervision	3.7	10.1	38.9%	1.8
Checklist+	3.8	11.5	**83.4%**	0.8

Human evaluation results on recipe

- Syntax: grammaticality
- Ingredient use: how well recipe adheres to ingredient list
- Follows goal: how well recipe accomplishes desired dish
- Surprisingly, Attention, EncDec and Checklist beat Truth in terms of grammar due to
  - noise in parsing the true recipes
  - neural models tend to generate shorter simpler texts

Model	Syntax	Ingredient use	Follows goal
Attention	4.47	3.02	3.47
EncDec	**4.58**	3.29	3.61
NN	4.22	3.02	3.36
NN-Swap	4.11	3.51	3.78
Checklist	**4.58**	3.80	3.94
Checklist+	4.39	**3.95**	**4.10**
Truth	4.39	4.03	4.34

Example qualitative analysis

EncDec	Checklist+
Cook broccoli according to package directions; drain. In medium saucepan, melt butter. Stir in flour, salt and pepper. Gradually add milk, stirring constantly. Cook over medium heat, stirring constantly, until mixture comes to a boil. Reduce heat to low. Cook, stirring constantly, until mixture boils and thickens, about 5 minutes. Stir in **milk**. Gradually heat to boiling; reduce heat. Cover; simmer 10 minutes or until heated through. Serve immediately.	Mix the turkey with the flour, salt, pepper, milk and lemon juice. Stir in the cheese. Add broccoli, mushrooms, nutmeg, and cheese. Mix well. Add the butter and mix well. Add the biscuits and mix well. Spread in a greased 13x9 inch baking pan. Bake in a 350f oven for 35 to 40 minutes, or until pick inserted in the center comes out clean. Serve hot.

ır, 5 eggs, 15 oreo cookies, 1/2 tsp vanilla, 1/2 tsp almond extract

	EncDec	Checklist+
add ggs. our hes r 1 wa- kle	Beat cream cheese and sugar together until light and fluffy. Add eggs, one at a time, beating well after each addition. Blend in vanilla. Sift flour, measure, and sift with **baking power** and **salt**. Add to creamed mixture alternately with **milk**. Fold in stiffly beaten **egg whites**. Pour into 9x13 pan. Bake at 350 degrees for 25 to 30 minutes.	Cream sugar and cream cheese until smooth. Add eggs, one at a time, beating well after each addition. Add vanilla, almond extract, and cookies. Mix well. Pour into greased and floured tube pan. Bake at 350 degrees for 30 minutes.

Conclusion

- RNNs (esp. GRU and LSTM) are well suited for natural language generation tasks
- Baseline RNN guarantees local coherence, while integration of agenda items (attention) guarantees global coverage
- Commonly used metrics (such as BLEU and METEOR) may not be a good measurement
  - Typically, human evaluation will be needed

Thank you!