Learning Neural Templates for Text Generation

EMNLP 2018

Sam Wiseman、Stuart M. Shieber、Alexander M. Rush

Harvard University

Xiachong Feng

Outline

1. Author
2. Overview
3. Motivation
4. Task
5. Semi-Markov Models
6. Neural HSMM Decoder
7. Experiment
8. Conclusion

Author

- **Sam Wiseman**
- Research Assistant Professor at **TTIC** *（丰田工业大学芝加哥分校）*
- Before TTIC
  - a PhD student in Computer Science at **Harvard**
  - a member of the **harvardnlp** group

Overview

- **Task: Generate textual descriptions of knowledge base records.**

Extract by hidden semi-markov model

Neural Template:

The ___ / ___ / …	is a / is an / is an expensive / …	___	providing / serving / offering / …	___
food / cuisine / foods / …	in the / with a / and has a / …	___	price range / price bracket / pricing / …	.	It's / It is / The place is / …
located in the / located near / near / …	___	.	Its customer rating is / Their customer rating is / Customers have rated it / …	___	.

neural template **learned by the system**

Given

Source Entity: Cotto

type[coffee shop], rating[3 out of 5], food[English], area[city centre], price[moderate], near[The Portland Arms]

Knowledge base records

Encoder-decoder

System Generation:

Cotto is a coffee shop serving English food in the moderate price range. It is located near The Portland Arms. Its customer rating is 3 out of 5.

template-like

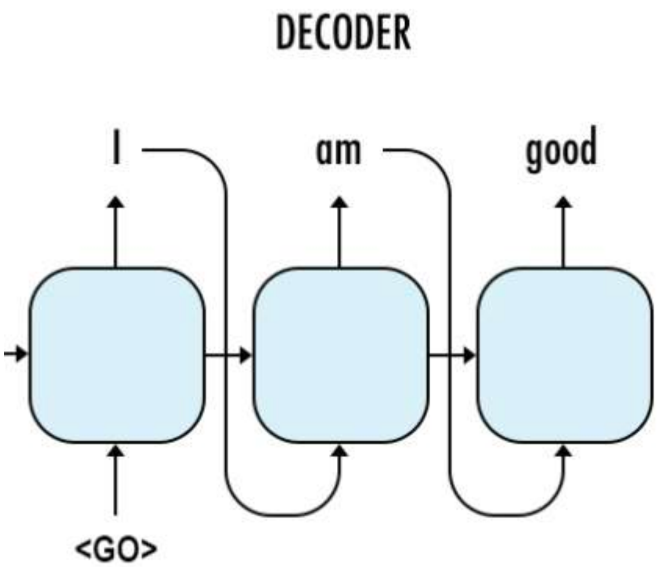
system generation

Motivation

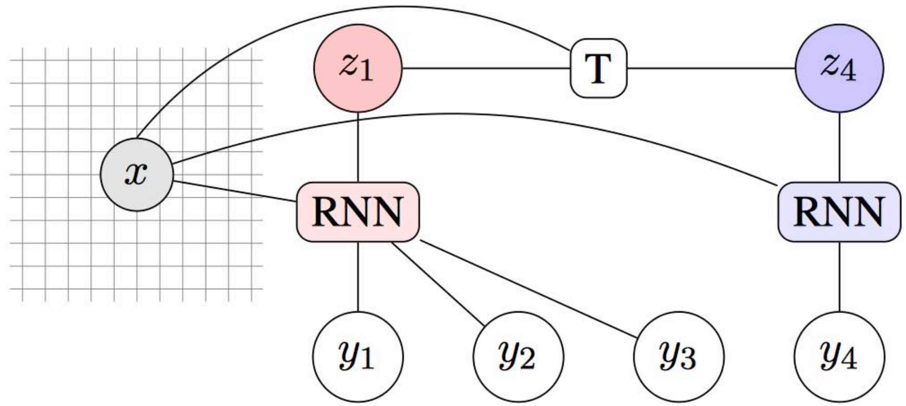
- Due to the **black-box** nature of generic encoder-decoder models
  - Uninterpretable
  - Difficult to control in terms of their phrasing or content.

- **Template-like** text generation
  - what to say
  - how to say

Motivation



Neural Decoder



HSMM Decoder

Continuous latent variable → **Discrete latent variable**

Segments are independent of each other given the corresponding latent variable and x.

Task

- **Task**: generating a textual description of a knowledge base or meaning representation.
- **Given**
  - A collection of records $x = \{r_1 \dots r_J\}$
    - Type: $(r.t)$
    - Entity: $(r.e)$
    - Value: $(r.m)$

Source Entity: Cotto

type[coffee shop], rating[3 out of 5], food[English], area[city centre], price[moderate], near[The Portland Arms]

- **Output**: *adequate* and *fluent* text description of x

$$\hat{y}_{1:T} = \hat{y}_1, \dots, \hat{y}_T$$

- **Dataset:**
  - E2E Dataset
  - WikiBio dataset

Dataset

Flat MR	NL reference
name[Loch Fyne], eatType[restaurant], food[French], priceRange[less than £20], familyFriendly[yes]	Loch Fyne is a family-friendly restaurant providing wine and cheese at a low cost. Loch Fyne is a French family friendly restaurant catering to a budget of below £20. Loch Fyne is a French restaurant with a family setting and perfect on the wallet.

reference text

E2E Dataset

Frederick Parker-Rhodes	
Born	21 November 1914 Newington, Yorkshire
Died	2 March 1987 (aged 72)
Residence	UK
Nationality	British
Fields	Mycology, Plant Pathology, Mathematics, Linguistics, Computer Science
Known for	Contributions to computational linguistics, combinatorial physics, bit-string physics, plant pathology, and mycology
Author abbrev. (botany)	Park.-Rhodes

reference text

Frederick Parker-Rhodes (21 March 1914 - 21 November 1987) was an English linguist, plant pathologist, computer scientist, mathematician, mystic, and mycologist.

WikiBio dataset

Semi-Markov Models

- A semi-Markov HMM is like an HMM except **each state can emit a sequence of observations**

- **HMM**
  - **Observed tokens :** $y_1 \cdots y_T$
  - **Latent state :** $z_t \in \{1, \ldots, K\}$

- **Semi-Markov models**
  - a length variable: $l_t \in \{1, \ldots, L\}$
    - the length of the current segment
  - a deterministic **binary** variable: f_t
    - whether a segment finishes at time t
    - **0-remain in same state**
    - **1-transition**

per-timestep *variables*

Semi-Markov Models

- **Joint-likelihood**

$$p(y, z, l, f \mid x; \theta) = \prod_{t=0}^{T-1} p(z_{t+1}, l_{t+1} \mid z_t, l_t, x)^{f_t} \times \prod_{t=1}^{T} p(y_{t-l_t+1:t} \mid z_t, l_t, x)^{f_t},$$

- **Assume**

$$p(z_{t+1}, l_{t+1} \mid z_t, l_t, x) \longrightarrow p(z_{t+1} \mid z_t, x) \times p(l_{t+1} \mid z_{t+1})$$

- **Final**
  - the probabilities of each **discrete state transition**

    $$\boxed{p(z_{t+1} \mid z_t, x)}$$

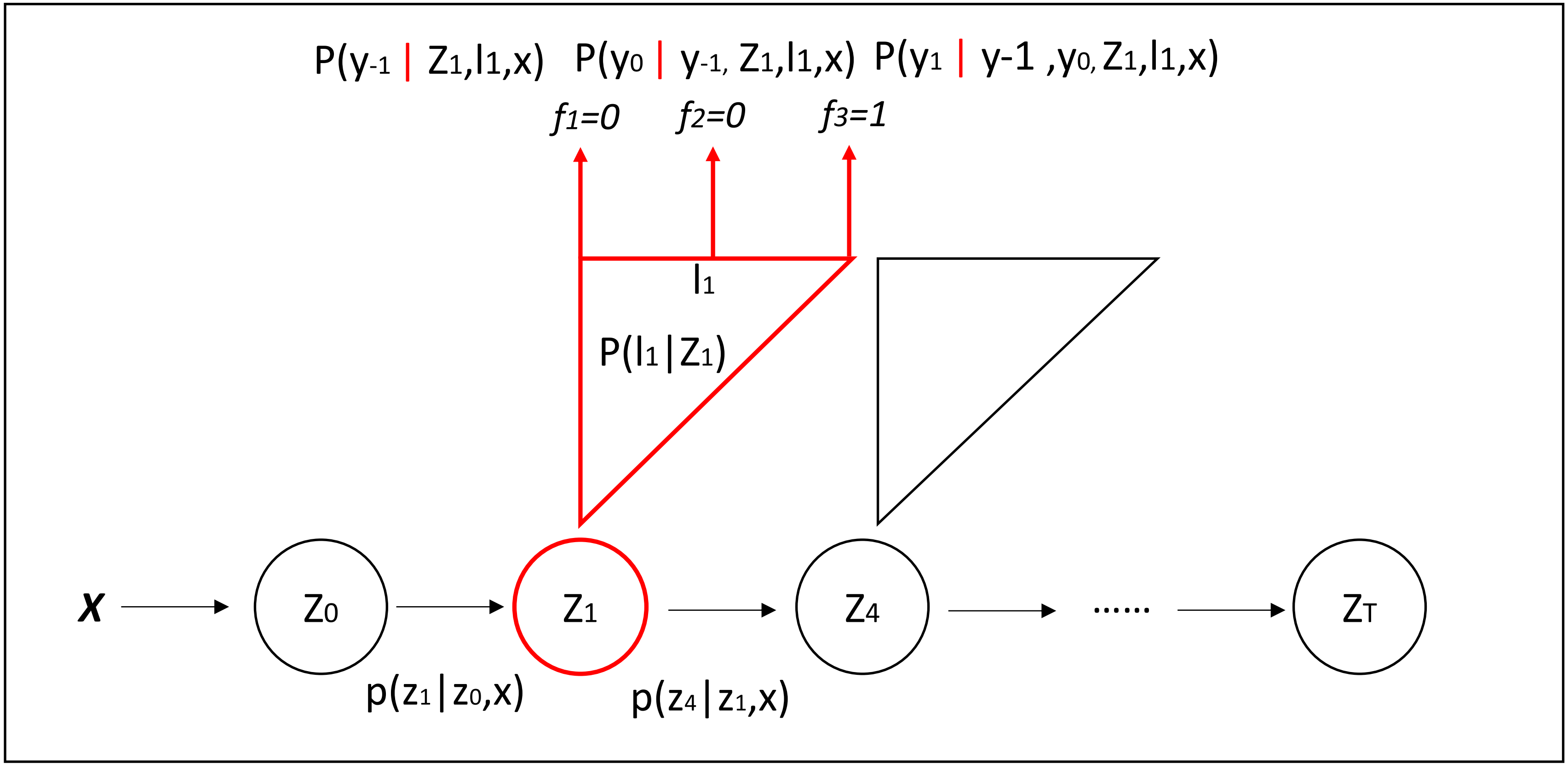
  - the probability of the **length of each segment** given its discrete state

    $$\boxed{p(l_{t+1} \mid z_{t+1})}$$

  - the probability of the **observations** in each segment, given its state and length.

    $$\boxed{p(y_{t-l_t+1:t} \mid z_t, l_t, x)}$$

Semi-Markov Models



$$p(y, z, l, f \mid x; \theta) = \prod_{t=0}^{T-1} p(z_{t+1}, l_{t+1} \mid z_t, l_t, x)^{f_t} \times \prod_{t=1}^{T} p(y_{t-l_t+1:t} \mid z_t, l_t, x)^{f_t},$$

Semi-Markov Models

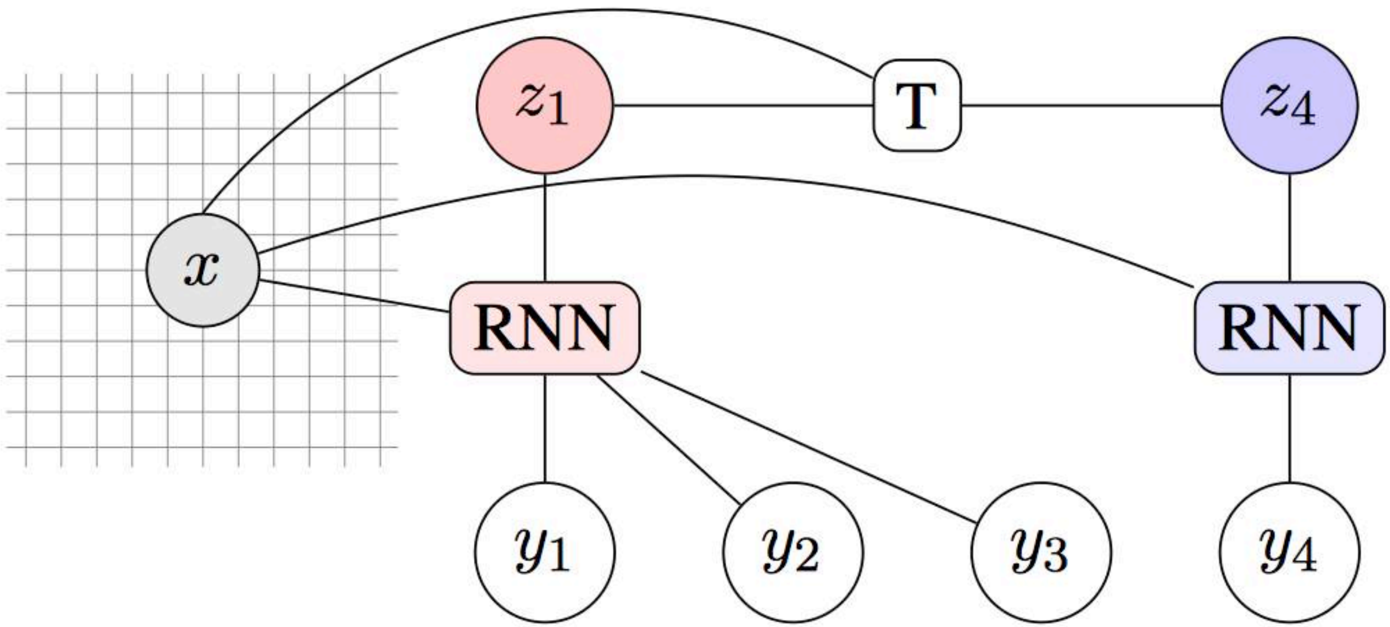
- **Given**
  - **HSMM （transition + emission）** *have learned*
- **Probability**
  - the probabilities of each **discrete state transition**

    $\boxed{p(z_{t+1} \mid z_t, x)}$
  - the probability of the **length of each segment** given its discrete state

    $\boxed{p(l_{t+1} \mid z_{t+1})}$
  - the probability of the **observations** in each segment, given its state and length.

    $\boxed{p(y_{t-l_t+1:t} \mid z_t, l_t, x)}$

A Neural HSMM Decoder

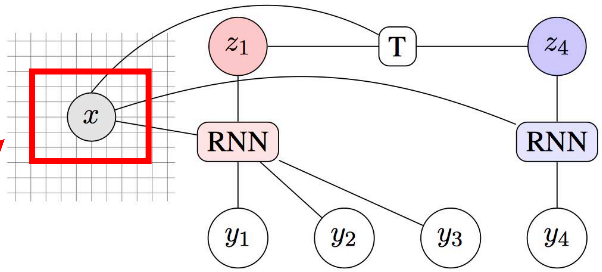


independently of the previous words

Parameterization

$\boldsymbol{r}_j \in \mathbb{R}^d$

- *real embedding of **record** $r_j \in x$*



$\boldsymbol{x}_a \in \mathbb{R}^d$

- *real embedding of the entire **knowledge base x***
- *obtained by max-pooling coordinate-wise over all the* $\boldsymbol{r}_j$

$\boldsymbol{x}_u \in \mathbb{R}^d$

- *representation of just the unique **types** of records*
- *the sum of the embeddings of the unique types appearing in x, plus a bias vector and followed by a ReLU nonlinearity.*

Transition & Length

- **Transition distribution**

K x K matrix

$$\boxed{p(z_{t+1}\,|\,z_t, x) \propto \boldsymbol{A}\boldsymbol{B} + \boldsymbol{C}(\boldsymbol{x}_u)\boldsymbol{D}(\boldsymbol{x}_u),}$$

$$\left\{\begin{array}{l} \boldsymbol{A} \in \mathbb{R}^{K\times m_1}, \boldsymbol{B} \in \mathbb{R}^{m_1\times K} \\ \boldsymbol{C} : \mathbb{R}^d \to \mathbb{R}^{K\times m_2} \\ \boldsymbol{D} : \mathbb{R}^d \to \mathbb{R}^{K\times m_2} \end{array}\right.$$

state embeddings

non-linear functions

- **Length distribution**
  - **Fix** all length probabilities $p(l_{t+1}\,|\,z_{t+1})$ to be **uniform** up to **a maximum length L**.

Emission Distribution

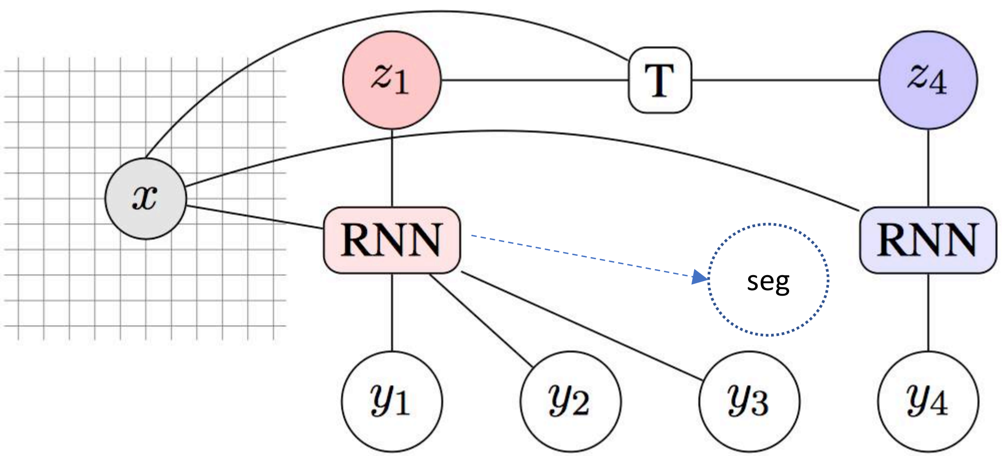
- Base this model on an **RNN decoder**
- Write a segment's probability as **a product over token-level probabilities**
- RNN decoder uses **attention** and **copy-attention**

</seg> is an end of segment token.

$$p(y_{t-l_t+1:t} \mid z_t = k, l_t = l, x) = \prod_{i=1}^{l_t} p(y_{t-l_t+i} \mid y_{t-l_t+1:t-l_t+i-1}, z_t = k, x) \times p(\text{</seg>} \mid y_{t-l_t+1:t}, z_t = k, x) \times \mathbf{1}_{\{l_t = l\}},$$

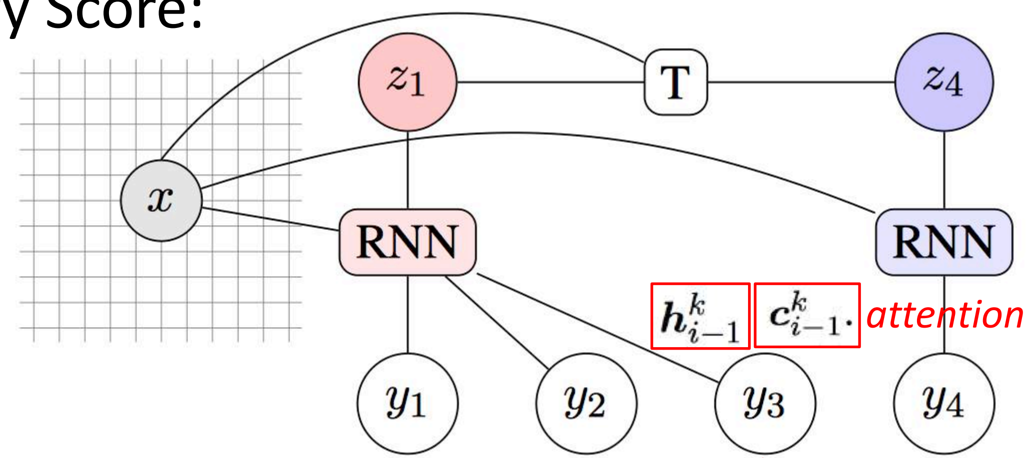
concatenating an embedding corresponding to the k'th latent state to the RNN's input

Each seg** initializing its hidden state with **x



Emission Distribution

Vocabulary Score:



$$\boldsymbol{v}_{i-1} = \boldsymbol{W} \tanh(\boldsymbol{g}_1^k \circ [\boldsymbol{h}_{i-1}^k, \boldsymbol{c}_{i-1}^k]),$$

Copy score*(For every r)*

$$\boxed{\rho_j = \boldsymbol{r}_j^{\mathsf{T}} \tanh(\boldsymbol{g}_2^k \circ \boldsymbol{h}_{i-1}^k),}$$

Final Score:

$$\boxed{\widetilde{\boldsymbol{v}}_{i-1} = \mathrm{softmax}([\boldsymbol{v}_{i-1}, \rho_1, \ldots, \rho_J]),}$$

Autoregressive Variant

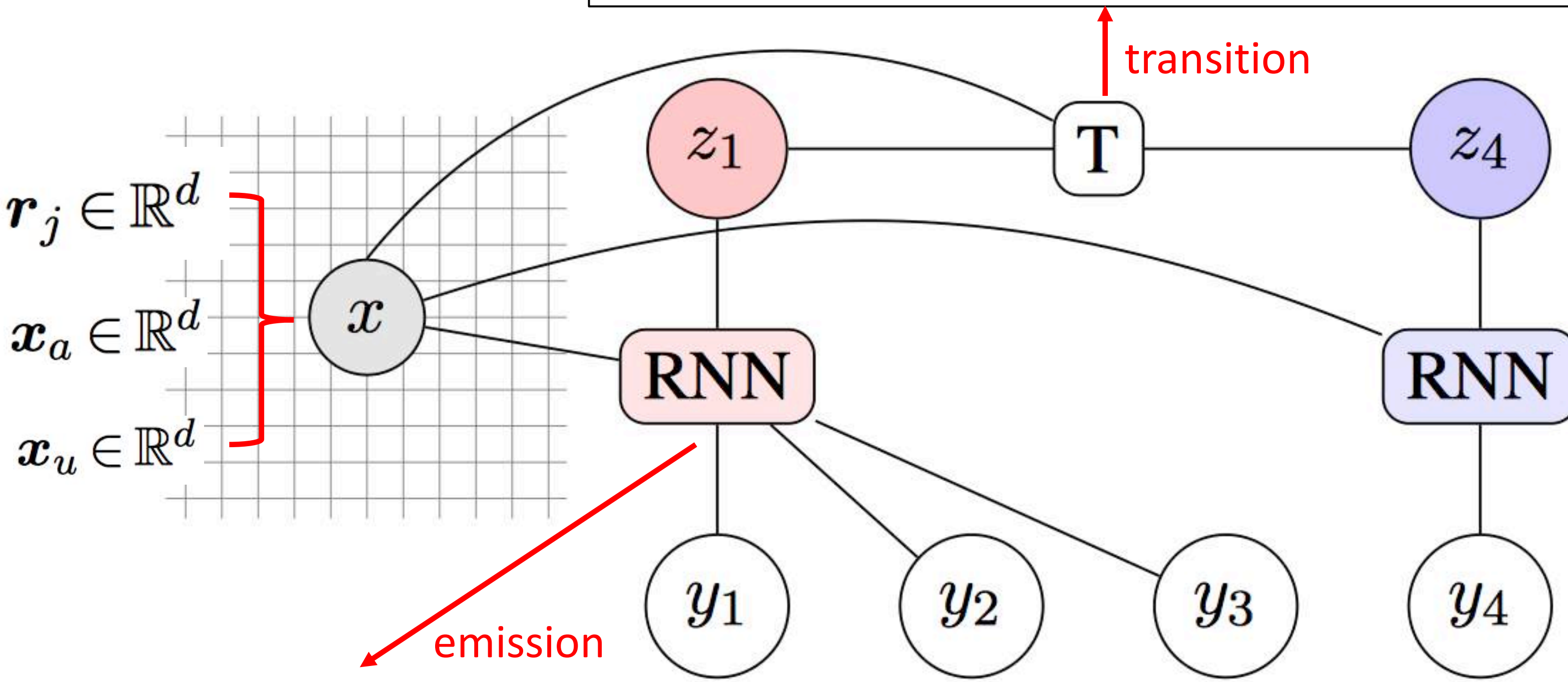
- allow **interdependence** between tokens (but not segments) by having each next-token distribution **depend on all the previously generated tokens**
- using **an additional RNN** run over all the preceding tokens.

$$p(y_{t-l_t+i} \mid \boxed{y_{t-l_t+1}}{}_{:t-l_t+i-1}, z_t = k, x)$$

$$p(y_{t-l_t+i} = w \mid \boxed{y_1}{}_{:t-l_t+i-1}, z_t = k, x)$$

Brief Summary

$$p(z_{t+1} \mid z_t, x) \propto \boldsymbol{A}\boldsymbol{B} + \boldsymbol{C}(\boldsymbol{x}_u)\boldsymbol{D}(\boldsymbol{x}_u),$$



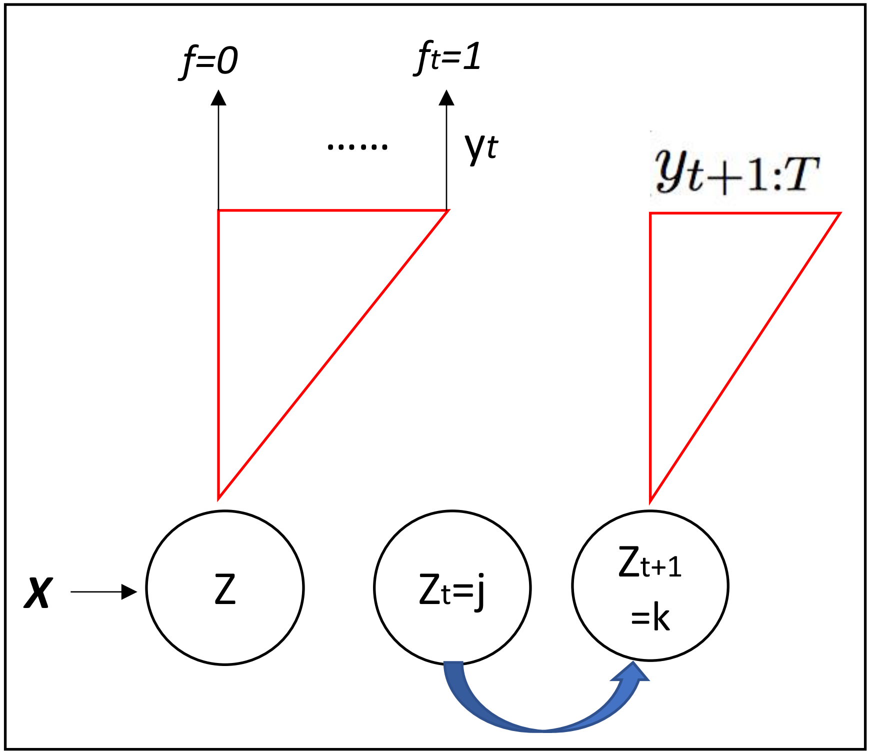
$$\begin{aligned} &p(y_{t-l_t+1:t} \mid z_t = k, l_t = l, x) = \\ &\quad \prod_{i=1}^{l_t} p(y_{t-l_t+i} \mid y_{t-l_t+1:t-l_t+i-1}, z_t = k, x) \\ &\quad \times p(</\text{seg}> \mid y_{t-l_t+1:t}, z_t = k, x) \times \mathbf{1}_{\{l_t = l\}}, \end{aligned}$$

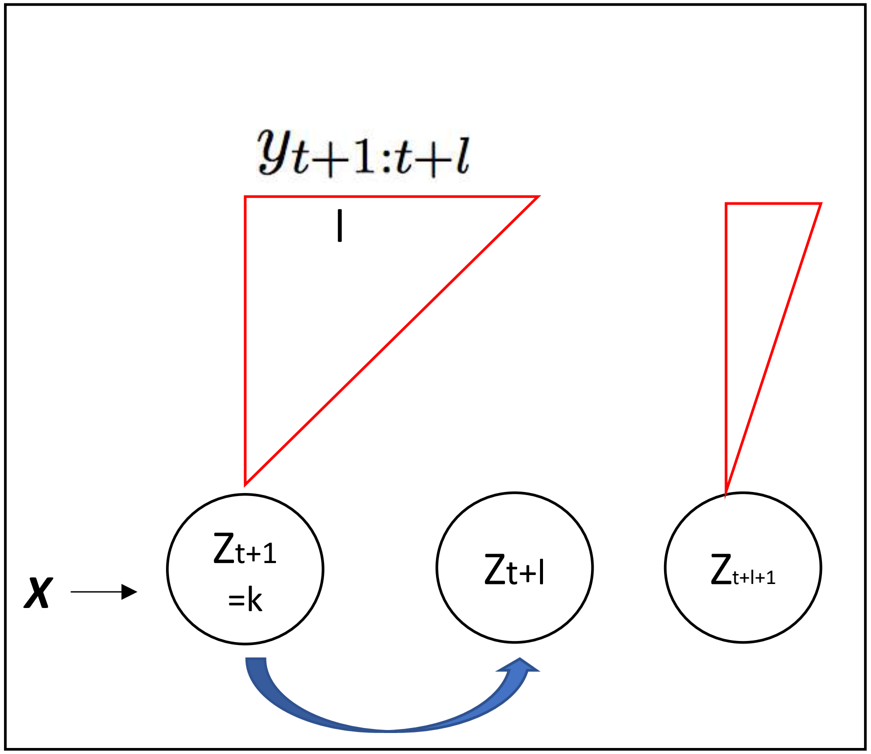
Learning

Backward algorithm

$$\beta_t(j) = p(y_{t+1:T} \mid z_t = j, f_t = 1, x)$$
$$= \sum_{k=1}^{K} \beta_t^*(k)\, p(z_{t+1} = k \mid z_t = j)$$

$$\beta_t^*(k) = p(y_{t+1:T} \mid z_{t+1} = k, f_t = 1, x)$$
$$= \sum_{l=1}^{L} \Big[\beta_{t+l}(k)\, p(l_{t+1} = l \mid z_{t+1} = k)$$
$$p(y_{t+1:t+l} \mid z_{t+1} = k, l_{t+1} = l)\Big],$$





Learning

$\beta_T(j) = 1.$ *Already the last time step*

$p(y \mid x) = \sum_{k=1}^{K} \beta_0^*(k)\, p(z_1 = k)$ *From start step use dynamic programming*

The final objective function

$$\boxed{\ln p(y \mid x; \theta) = \ln \sum_{k=1}^{K} \beta_0^*(k)\, p(z_1 = k).}$$

What can we do now？

After training（We get HSMM）, we could simply condition on a **new database** and generate with **beam search**, as is standard with **encoder-decoder** models.

But what do we mean template-like？

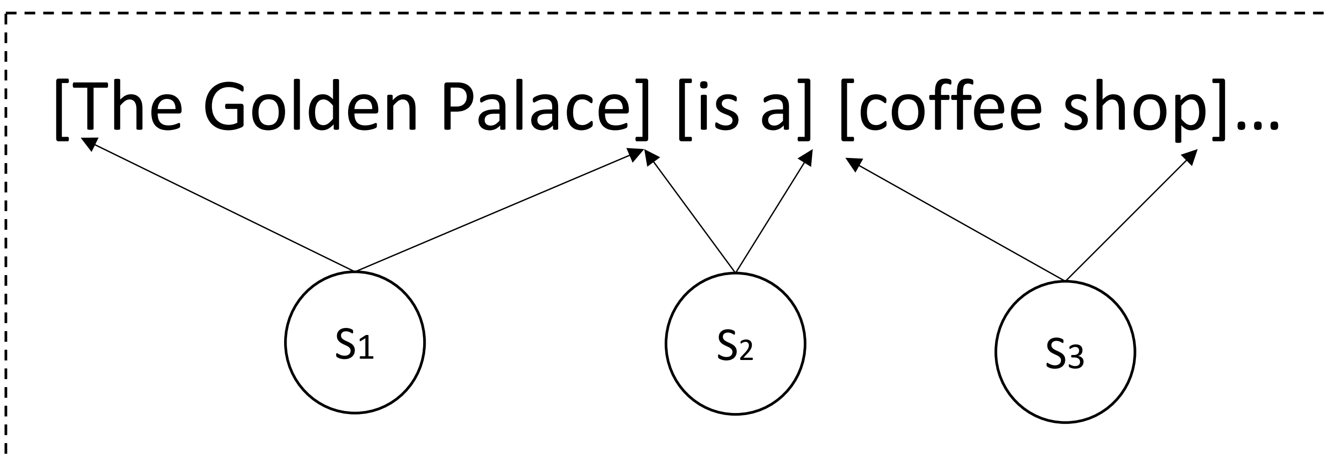
HSMM-Decoding

Given

- ***HSMM*** ***we have already learned***
- ***Data*** ***which describes knowledge***

Goal

- ***Find the best hidden states sequence***

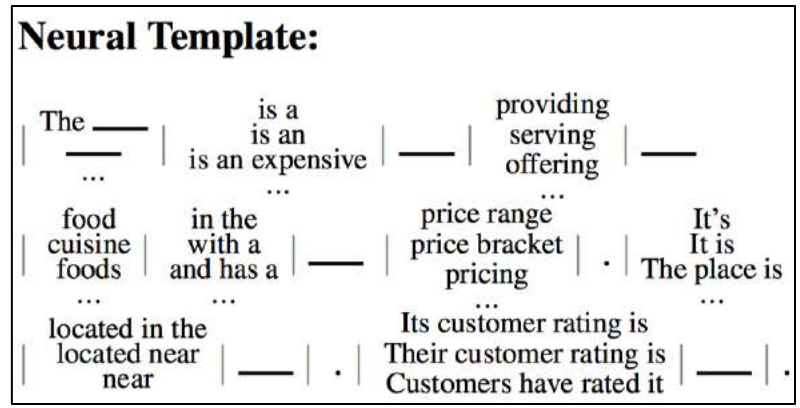


Extracting Templates

- **Templates**: sequences of hidden states
- Each "template" $z^{(i)}$ consists of a sequence of latent states

[The Golden Palace]$_{55}$ [is a]$_{59}$ [coffee shop]$_{12}$ [providing]$_{3}$ [Indian]$_{50}$ [food]$_{1}$ [in the]$_{17}$ [£20-25]$_{26}$ [price range]$_{16}$ [.]$_{2}$ [It is]$_{8}$ [located in the]$_{25}$ [riverside]$_{40}$ [.]$_{53}$ [Its customer rating is]$_{19}$ [high]$_{23}$ [.]$_{2}$

Figure 4: A sample Viterbi segmentation of a training text; subscripted numbers indicate the corresponding latent state. From this we can extract a template with $S = 17$ segments; compare with the template used at the bottom of Figure 1.



template

visualization

discrete states are replaced by the phrases they frequently generate in the training data.

$$\hat{y}^{(i)} = \arg\max_{y'} p(y', z^{(i)} \mid x),$$

Experiment

The E2E task

	BLEU	NIST	ROUGE	CIDEr	METEOR
			Validation		
D&J	69.25	8.48	72.57	2.40	47.03
SUB	43.71	6.72	55.35	1.41	37.87
NTemp	66.50	7.87	69.24	2.20	44.45
NTemp+AR	67.12	7.98	69.55	2.30	43.21
			Test		
D&J	65.93	8.59	68.50	2.23	44.83
SUB	43.78	6.88	54.64	1.39	37.35
NTemp	58.88	7.54	65.71	2.02	41.21
NTemp+AR	59.80	7.56	65.01	1.95	38.75

The WikiBio

	BLEU	NIST	ROUGE-4
Template KN †	19.8	5.19	10.7
NNLM (field) †	33.4	7.52	23.9
NNLM (field & word) †	34.7	7.98	25.8
NTemp	34.2	7.94	35.9
NTemp+AR	34.8	7.59	38.6
Seq2seq (Liu et al., 2018)	43.65	-	40.32

- the templated baselines underperform neural models
- our proposed model is fairly competitive with neural models, and sometimes even outperforms them.

Experiment-Controllable

Travellers Rest Beefeater
name[Travellers Rest Beefeater], customerRating[3 out of 5], area[riverside], near[Raja Indian Cuisine]
1. [Travellers Rest Beefeater]$_{55}$ [is a]$_{59}$ [3 star]$_{43}$ [restaurant]$_{11}$ [located near]$_{25}$ [Raja Indian Cuisine]$_{40}$ [.]$_{53}$ 2. [Near]$_{31}$ [riverside]$_{29}$ [,]$_{44}$ [Travellers Rest Beefeater]$_{55}$ [serves]$_{3}$ [3 star]$_{50}$ [food]$_{1}$ [.]$_{2}$ 3. [Travellers Rest Beefeater]$_{55}$ [is a]$_{59}$ [restaurant]$_{12}$ [providing]$_{3}$ [riverside]$_{50}$ [food]$_{1}$ [and has a]$_{17}$ [3 out of 5]$_{26}$ [customer rating]$_{16}$ [.]$_{2}$ [It is]$_{8}$ [near]$_{25}$ [Raja Indian Cuisine]$_{40}$ [.]$_{53}$ 4. [Travellers Rest Beefeater]$_{55}$ [is a]$_{59}$ [place to eat]$_{12}$ [located near]$_{25}$ [Raja Indian Cuisine]$_{40}$ [.]$_{53}$ 5. [Travellers Rest Beefeater]$_{55}$ [is a]$_{59}$ [3 out of 5]$_{5}$ [rated]$_{32}$ [riverside]$_{43}$ [restaurant]$_{11}$ [near]$_{25}$ [Raja Indian Cuisine]$_{40}$ [.]$_{53}$

Experiment-Interpretable

kenny warren

name: kenny warren, **birth date:** 1 april 1946, **birth name:** kenneth warren deutscher, **birth place:** brooklyn, new york, **occupation:** ventriloquist, comedian, author, **notable work:** book - the revival of ventriloquism in america

1. [kenneth warren deutscher]$_{132}$ [( ]$_{75}$ [born]$_{89}$ [april 1, 1946]$_{101}$ [ )]$_{67}$ [is an american]$_{82}$ [author]$_{20}$ [and]$_{1}$ [ventriloquist and comedian]$_{69}$ [.]$_{88}$
2. [kenneth warren deutscher]$_{132}$ [( ]$_{75}$ [born]$_{89}$ [april 1, 1946]$_{101}$ [ )]$_{67}$ [is an american]$_{82}$ [author]$_{20}$ [best known for his]$_{95}$ [the revival of ventriloquism]$_{96}$ [.]$_{88}$
3. [kenneth warren]$_{16}$ ["kenny" warren]$_{117}$ [( ]$_{75}$ [born]$_{89}$ [april 1, 1946]$_{101}$ [ )]$_{67}$ [is an american]$_{127}$ [ventriloquist, comedian]$_{28}$ [.]$_{133}$
4. [kenneth warren]$_{16}$ ["kenny" warren]$_{117}$ [( ]$_{75}$ [born]$_{89}$ [april 1, 1946]$_{101}$ [ )]$_{67}$ [is a]$_{104}$ [new york]$_{98}$ [author]$_{20}$ [.]$_{133}$
5. [kenneth warren deutscher]$_{42}$ [is an american]$_{82}$ [ventriloquist, comedian]$_{118}$ [based in]$_{15}$ [brooklyn, new york]$_{84}$ [.]$_{88}$

particular discrete states correspond in a consistent way to particular pieces of information, allowing us to align states with particular field types. For instance, birth names have the same hidden state (132), as do names (117), nationalities (82), birth dates (101), and occupations (20).

Thanks!