Cluster-based Beam Search for Pointer-Generator Chatbot Grounded by Knowledge

Yik-Cheung Tam, Jiachen Ding[2], Cheng Niu, Jie Zhou
WeChat AI - Pattern Recognition Center, Tencent Inc
Shanghai Jiaotong University[2]
{wilsontam,niucheng,withtomzhou}@tencent.com
ding222@sjtu.edu.cn[2]

Agenda

- Introduction
- Proposed System
  - Model
  - Decoding Strategies
- Evaluation Results
- Conclusions

Introduction

- DSTC7 Track 2 challenge: Design a chatbot that is:
    - Grounded by unstructured external knowledge (facts)
    - Conversational-history aware
    - End-to-end trainable
- Diverse response generation
    - Relevant
    - Interesting

Issues

- How to integrate conversational history (H) and external knowledge (F)?

- How to generate diverse responses (R)?

Related Work

- Seq-to-Seq with copy mechanism
  - Chit-Chat: Gu 2016, See 2017
  - KB: Eric and Manning 2017
- Response diversity
  - MMI: Li 2016, Zhang 2018, Baheti 2018
  - Latent variables: Zhou 2017, Su 2018
- Beam search decoding
  - Diversity objective: Vijayakumar 2018
- Knowledge-grounded
  - Memory network: Ghazvininejad 2018, Madotto 2018

Many more papers..

Contributions

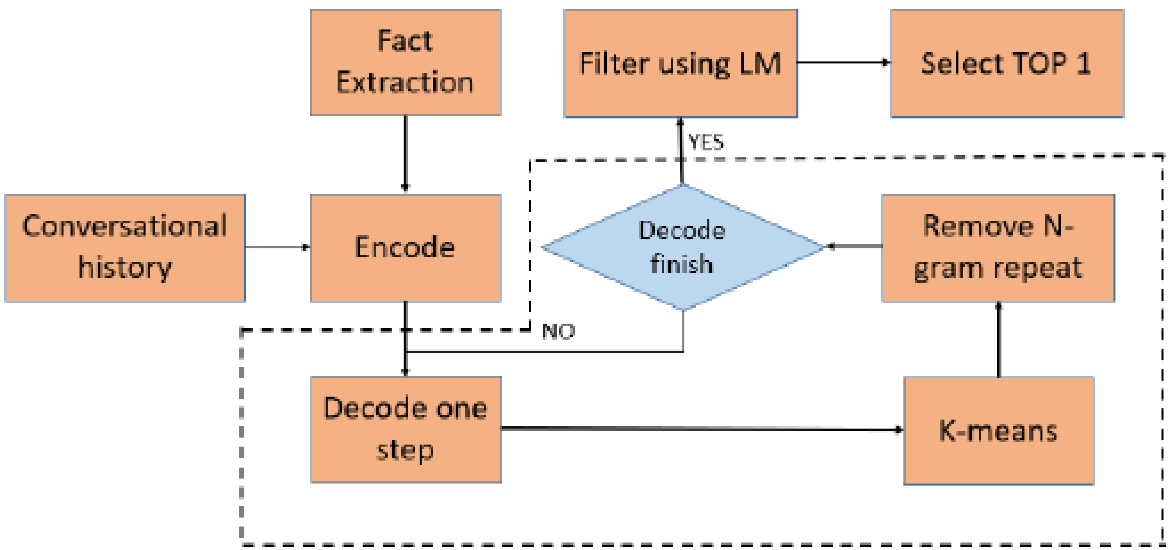
Model:

- Enable pointer generator to copy from history and facts

Decoding Strategies:

- Cluster-based beam search
- Safe response filtering using LM

Overall System Diagram

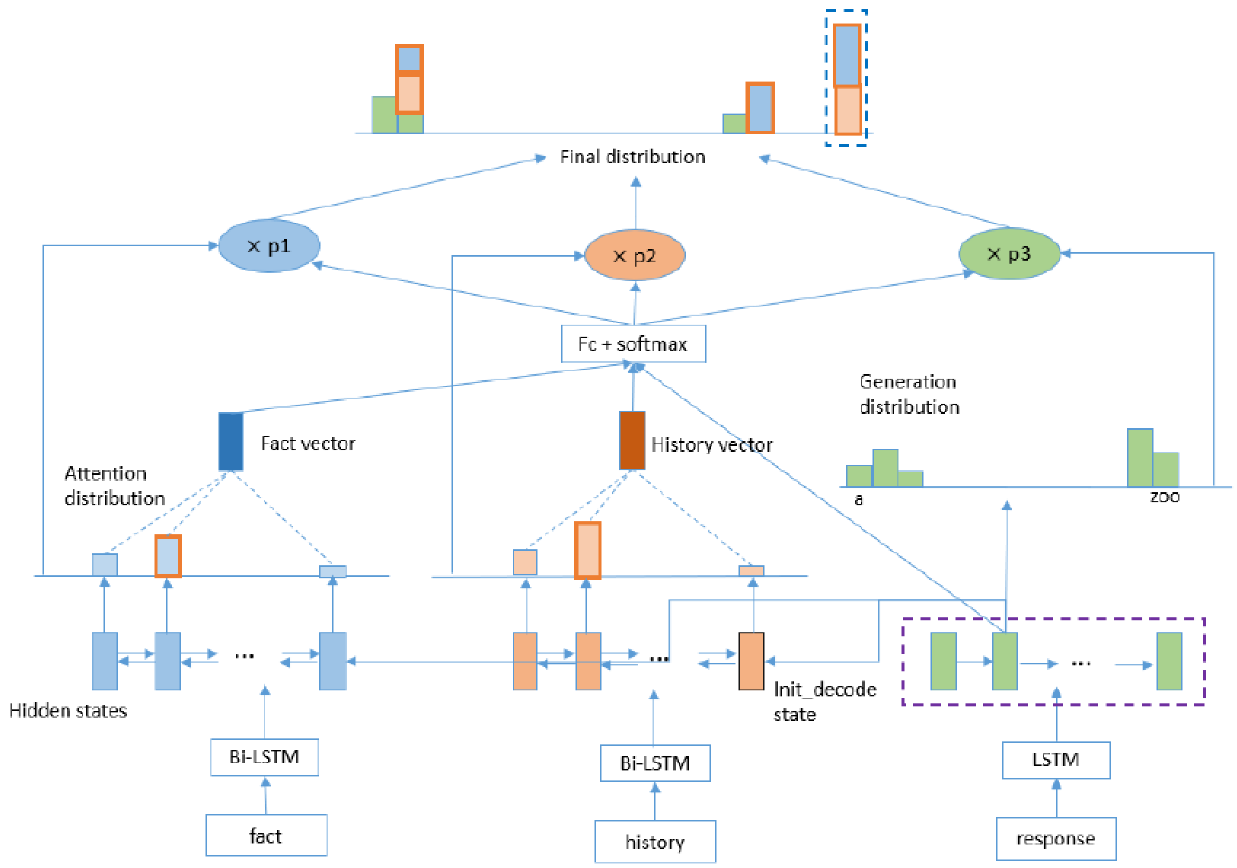


Pointer-Generator

- A Seq-to-Seq model with copy-mechanism [See et.al. 2017]
  - OOV can be copied to output

- At each decoding step, our model chooses to:
  1. Generate a token based on vocabulary V
  2. Copy a token from history
  3. Copy a token from facts

One decoding step: Pr(word|F,H)



Attention Mechanism

- BiLSTM history vectors: $\{h_1^H, h_2^H, ..., h_i^H, ..., h_L^H\}$
- BiLSTM fact vectors: $\{h_1^F, h_2^F, ..., h_j^F, ..., h_T^F\}$

Attention over history vectors at decoding step t

$$e_{ti}^H = v_H^T \cdot \tanh(W_h^H \cdot h_i^H + W_r^H \cdot h_t^R + b^H)$$

$$\alpha_{ti}^H = Softmax(e_{ti}^H)$$

$$h_t^{H*} = \sum_{i=1}^{L} \alpha_{ti}^H h_i^H$$

Attention over fact vectors at decoding step t

$$e_{tj}^F = v_H^F \cdot \tanh(W_h^F \cdot h_j^F + W_r^F \cdot h_t^R + b^F)$$

$$\alpha_{tj}^F = Softmax(e_{tj}^F)$$

$$h_t^{F*} = \sum_{j=1}^{T} \alpha_{tj}^F h_j^F$$

Mode prediction

- {Generate, Copy from fact, Copy from history}
- Concatenate all available features at decoder step t:
  - Attended fact vector
  - Attended history vector
  - Decoder hidden state
  - Last input word embedding
- Feed-Forward followed by Softmax over modes:

$$Pr(mode = m|t, H, F) =$$
$$Softmax(FF(h_t^{F*} \oplus h_t^{H*} \oplus h_t^{R} \oplus x_t))$$

Output word distribution

- Linearly interpolate distributions from 3 modes:

$$Pr(w|t,H,F) = \sum_{m=1}^{3} Pr(m|t,H,F) \cdot Pr_m(w|t,H,F)$$

- End-to-end trainable with cross-entropy

Decoding Issues

- Motivation: Safe responses are common in responses generated by Seq-to-Seq models
  - "This is the best thing I have ever seen"
- Observation from beam search:
  - Many responses are similar
    - "This is the best thing I have ever seen"
    - "This is the coolest thing I have ever seen"
  - Under a fixed beam budget, this is inefficient

Proposal: Cluster-based beam search

- Cluster similar partial hypotheses
- Prune per cluster

Beam Search with K-means

- Averaged word embedding to represent a candidate
- Apply K-means over extended candidates
- Prune candidates per cluster using Beam Size / K
- Remove repeated N-grams
- Filter out final meaningless candidates using LM

Algorithm 1: Beam search with K-means

Input: Beam size BS, Candidates C initialized with start symbol
Output: Final response rsp
Data: Language model threshold lm_{th}

while *Number of completed hypothesis does not reach* BS *or maximum decoding step is not reached* **do**
    for i *in* BS **do**
        $tmpHyps$=Top-N(Extend($C[i]$), $BS \times 2$);
        Remove hyps in $tmpHyps$ with repeated N-grams or UNK;
        Save $tmpHyps$ to extended candidates;
    end
    Perform K-means over extended candidates;
    for *candidates in each cluster* **do**
        Sort candidates by partial log-prob scores;
        Choose top BS/K candidates;
        Put candidates with end symbol in R;
        Put incomplete candidates in C_{new};
    end
    $C \leftarrow C_{new}$
end
Sort R according to log-prob scores;
for hyp *in* R **do**
    if $score_{lm}(hyp) < lm_{th}$ **then**
        $rsp \leftarrow hyp$;
        break;
    end
end

Experimental Setup

- Glove word embedding
- Decode validation set to obtain N-best responses to train an LM for safe response filtering
- Single system

Table 2: Hyper-parameter settings.

Name	Value
Vocabulary size	100,000
Word embedding size	300
LSTM hidden size	150
Batch size	128
Beam size (BS)	50
K-means clusters	10
N-gram repeat order	2
LM threshold (lm_{th})	-35
Learning rate for Adam	0.0005
Maximum gradient norm	2

Table 1: Dataset statistics for DSTC7 Track 2.

	Train set	Validation set	Test set
Samples	1,408,951	4,542	13,108

Official Evaluation Results

- Achieved best results on NIST-4, BLEU-4, Meteor
  - Cluster-based beam search help

Table 3: Automatic evaluation results. A total of 2208 test samples were evaluated. Best non-baseline results are marked in **bold**.

Name	NIST-4	BLEU-4	Meteor	Entropy-4	Div-1	Div-2	Avg len
Baseline (constant)	0.184	2.87 %	7.48 %	1.609	0.000	0.000	8
Baseline (random)	1.637	0.86%	5.91%	10.467	0.160	0.647	19.192
Baseline (seq2seq)	0.916	1.82%	6.96%	5.962	0.014	0.048	10.604
Team C/E	1.515	1.32%	6.43%	7.639	0.053	0.171	12.674
Team G	2.040	1.05%	7.48%	**10.057**	0.108	**0.449**	22.336
Our system w/ K-means	**2.523**	1.83%	**8.07**%	9.030	0.109	0.325	15.133
Our system w/o K-means	1.771	**1.94**%	7.64%	8.194	0.094	0.267	12.770
Human	2.650	3.13%	8.31%	10.445	0.167	0.670	18.757

Official Human Evaluation Results

- 5-level judgement, 3 judges
- Achieved best scores on "Interest and Informativeness" & Overall

Table 4: Human evaluation results. Best non-baseline results are marked in **bold**.

	Relevance		Interest		Overall	
Model	Mean Score	95% CI	Mean Score	95% CI	Mean Score	95% CI
Baseline (constant)	2.60	(2.560, 2.644)	2.32	(2.281, 2.364)	2.46	(2.424, 2.500)
Baseline (random)	2.32	(2.269, 2.371)	2.35	(2.303, 2.401)	2.34	(2.288, 2.384)
Baseline (seq2seq)	2.91	(2.858, 2.963)	2.68	(2.632, 2.730)	2.80	(2.748, 2.844)
Team C/E (2nd place)	**3.05**	**(3.009, 3.093)**	2.77	(2.735, 2.812)	2.91	(2.875, 2.950)
Team G (3rd place)	2.82	(2.771, 2.870)	2.57	(2.525, 2.619)	2.70	(2.650, 2.742)
Our system (1st place)	2.99	(2.938, 3.042)	**2.87**	**(2.822, 2.922)**	**2.93**	**(2.882, 2.979)**
Human	3.61	(3.554, 3.658)	3.49	(3.434, 3.539)	3.55	(3.497, 3.596)

Case Study

- Response with K-means beam search is better

Conversational history	til that in greek mythology zeus thought that the earth is overpopulated and trojan war was instigated to depopulate the earth .
Relevance Interest	4.67 4.67
w/ K-means	james cameron wrote a book about the origins of war on this .
w/o K-means	i’ve never heard of this , but i didn’t know it was aliens .

Bad case

- Low relevance

Conversational history	til : chinese philanthropist , run run shaw (1907-2014) gave $ 1 + / - billion to fund education , aiding over 6000 construction projects . because so many buildings (5000 +) bear his name , yifu , it is often confused as a generic name .
Relevance Interest	1.33 3.00
w/ K-means	james cameron is the best name ever . i'm pretty sure he voiced the tianyi of tesla .
w/o K-means	he's a chinese restaurant . it's his name !

Conclusions

- Proposed an end-to-end trainable system for submission
- Achieved the 1st place in the competition
- Response quality requires further improvement
  - Relevance
  - Interest
  - Response consistency

Thank you!

WeChat AI - Pattern Recognition Center, Tencent Inc