Incorporating External Knowledge into Dialogue System

Jiazhan Feng

Mar 22nd, Peking University

Response Ranking with Deep Matching Networks and External Knowledge in Information-seeking Conversation Systems

Liu Yang[1] Minghui Qiu[2] Chen Qu [1] Jiafeng Guo[3] Yongfeng Zhang[4] W. Bruce Croft[1]
Jun Huang[2] Haiqing Chen[2]
[1] Center for Intelligent Information Retrieval, University of Massachusetts Amherst [2] Alibaba Group
[3] Institute of Computing Technology, Chinese Academy of Sciences [4] Dept. of Computer Science, Rutgers University
{lyang,chenqu,croft}@cs.umass.edu,{minghui.qmh,huangjun.hj,haiqing.chenhq}@alibaba-inc.com
guojiafeng@ict.ac.cn,yongfeng.zhang@rutgers.edu

[2018-SIGIR]

Motivation

- Much less attention has been paid to information oriented conversations (i.e. information-seeking conversations != task oriented conversations as they have clear goals like ordering or booking).

- Lack of modeling external knowledge beyond the dialogue utterances. (main)

Framework

- *Information retrieval (IR) module:*
  - obtain the information seeking conversation data $\mathcal{D}$ and external QA text collection $\mathcal{E}$
  - retrieve a small relevant set of QA pairs $\mathcal{P}$ from $\mathcal{E}$ with the response candidate $\mathcal{R}$ as the queries
- *External knowledge extraction (KE) module:*
  - extract useful information from $\mathcal{P}$ like term distributions or term co-occurrence matrices as external knowledge
- *Deep matching network (DMN) module:*
  - predict the matching score *f* of a context (consists of several utterances *u*) and response candidate *r* pair with extracted external knowledge

Deep Matching Networks with Pseudo-Relevance Feedback (DMN-PRF)

PRF: Obtain top-ranked documents from the initial retrieval as feedback to enhance the query representation. (as response candidates might be very short)

1. *Relevant QA Posts Retrieval:*
   - adopt different QA text collections for different conversation data (e.g. Stack Overflow data for MSDialog, AskUbuntu for UDC)
   - use response candidate as query to retrieve top P QA posts with BM25 as $\mathcal{P}$ (P=10, in this paper)

Update Version of TF-IDF

2. *Candidate Response Expansion:*
   - given $\mathcal{P}$, compute a language model θ (QA post = (title, body), in PRF we use body only)
   - extract most frequent *W* terms from θ (*W* = 10, in this paper)
   - append them at the end of response candidate *r*

3. *Interaction Matching Matrix:*
   - Compute two matrices, M_1 , M_2

$$m_{1,i,j} = \mathbf{e}_{r,i}^T \cdot \mathbf{e}_{u,j}.$$ ⟵ word embedding

$$m_{2,i,j} = \mathbf{h}_{r,i}^T \cdot \mathbf{h}_{u,j}.$$ ⟵ hidden states of Bi-GRU

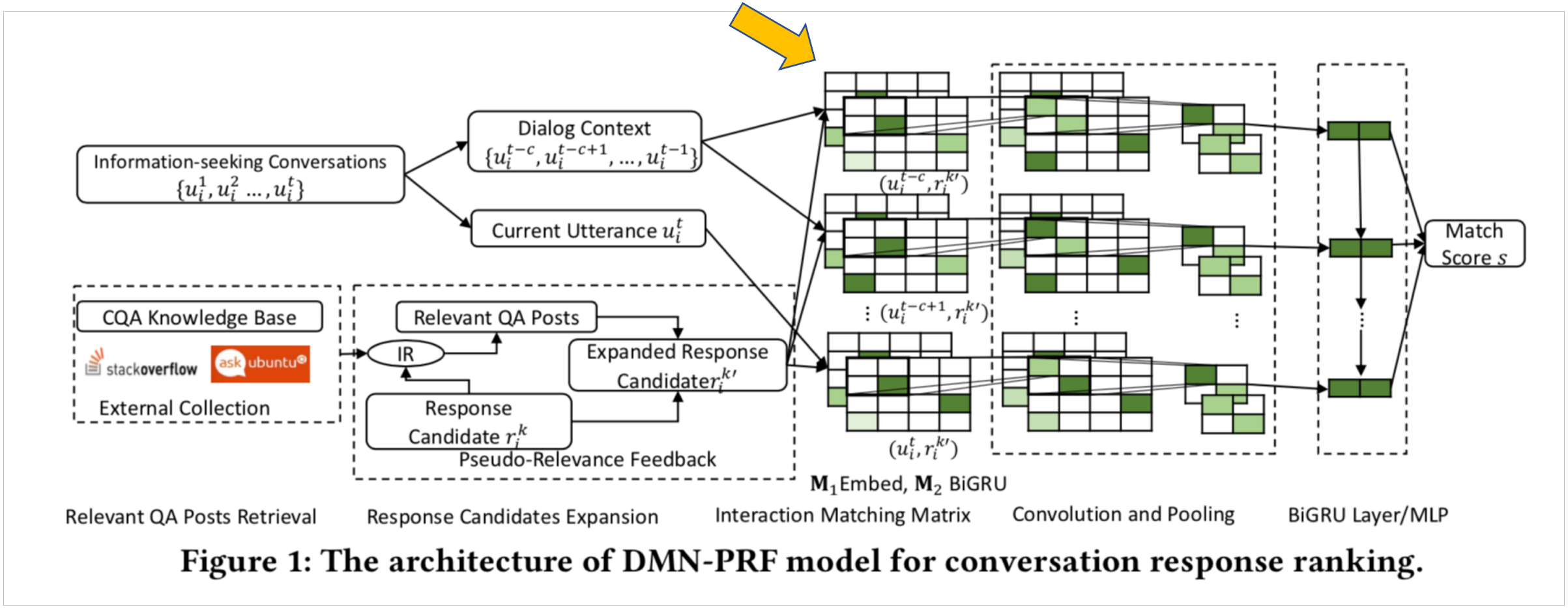
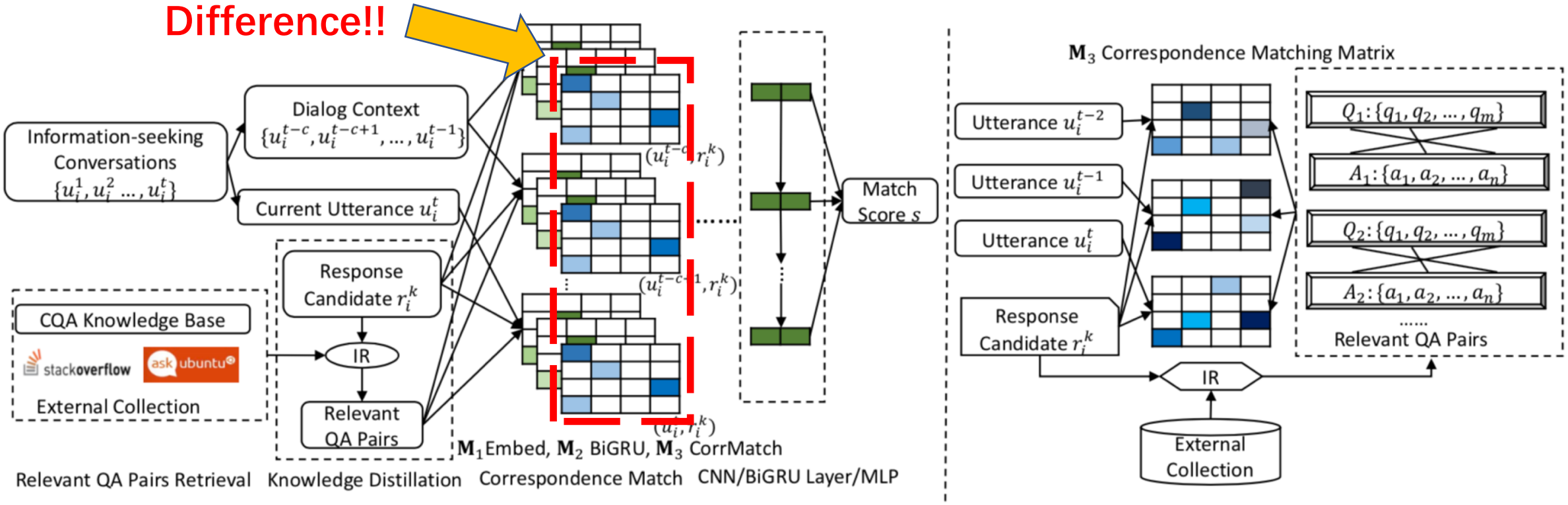


Figure 1: The architecture of DMN-PRF model for conversation response ranking.

Deep Matching Networks with QA Correspondence Knowledge Distillation (DMN-KD)

Correspondence matching knowledge: capture relationships such as *"(Problem Descriptions, Solutions)", "(Symptoms, Causes)", "(Information Request, Answers)"* etc. in the top ranked relevant QA pair set ($\mathcal{P}$ in this paper).



Correspondence matching knowledge matrix:

- Positive Pointwise Mutual Information (PPMI) of words of r_i^k and u_i^t in retrieved QA pair set {Q, A}.

$$
\begin{aligned}
m_{3,i,j} &= PPMI(w_{r,i}, w_{u,j} | \{\mathcal{Q}, \mathcal{A}\}) \\
&= \max(0, \log \frac{\sum_{p'=1}^{P} p(w_{r,i} \in \mathrm{A}_{p'}, w_{u,j} \in \mathrm{Q}_{p'} | \mathrm{Q}_{p'}, \mathrm{A}_{p'})}{p(w_{r,i} | \mathcal{A}) \cdot p(w_{u,j} | \mathcal{Q})})
\end{aligned}
$$

5. Training:

$$
\mathcal{L}(\mathcal{D}, \mathcal{E}; \Theta) = \sum_{i=1}^{I} \max(0, \epsilon - f(\mathcal{U}_i, r_i^{k+}) + f(\mathcal{U}_i, r_i^{k-})) + \lambda ||\Theta||_2^2
$$

Experiment

- Dataset (External QA Knowledge):
  1. Ubuntu Dialog Corpus (AskUbuntu)
  2. MSDialog (Stack Overflow data)
  3. AliMe Data (Clicked logs)

Table 5: Comparison of different models over Ubuntu Dialog Corpus (UDC), MSDialog, and AliMe data sets. Numbers in bold font mean the result is better compared with the best baseline. ‡ means statistically significant difference over the best baseline with $p < 0.05$ measured by the Student's paired t-test.

Data	UDC				MSDialog				AliMe			
Methods	MAP	Recall@5	Recall@1	Recall@2	MAP	Recall@5	Recall@1	Recall@2	MAP	Recall@5	Recall@1	Recall@2
BM25	0.6504	0.8206	0.5138	0.6439	0.4387	0.6329	0.2626	0.3933	0.6392	0.6407	0.2371	0.4204
BM25-PRF	0.6620	0.8292	0.5289	0.6554	0.4419	0.6423	0.2652	0.3970	0.6412	0.6510	0.2454	0.4209
ARC-II	0.6855	0.8978	0.5350	0.6959	0.5398	0.8662	0.3189	0.5413	0.7306	0.6595	0.2236	0.3671
MV-LSTM	0.6611	0.8936	0.4973	0.6733	0.5059	0.8516	0.2768	0.5000	0.7734	0.7017	0.2480	0.4105
DRMM	0.6749	0.8776	0.5287	0.6773	0.5704	0.9003	0.3507	0.5854	0.7165	0.6575	0.2212	0.3616
Duet	0.5692	0.8272	0.4756	0.5592	0.5158	0.8481	0.2934	0.5046	0.7651	0.6870	0.2433	0.4088
SMN	0.7327	0.9273	0.5948	0.7523	0.6188	0.8374	0.4529	0.6195	0.8145	0.7271	0.2881	0.4680
DMN	0.7363	0.9196	0.6056	0.7509	0.6415	0.9155	0.4521	0.6673	0.7833	0.7629	0.3568	0.5012
DMN-KD	**0.7655**‡	**0.9351**‡	**0.6443**‡	**0.7841**‡	**0.6728**‡	**0.9304**‡	**0.4908**‡	**0.7089**‡	**0.8323**	**0.7631**	**0.3596**‡	**0.5122**‡
DMN-PRF	**0.7719**‡	**0.9343**‡	**0.6552**‡	**0.7893**‡	**0.6792**‡	**0.9356**‡	**0.5021**‡	**0.7122**‡	**0.8435**‡	**0.7701** ‡	**0.3601** ‡	**0.5323** ‡

Model	Data Change	UDC MAP	UDC Recall@5	UDC Recall@1	UDC Recall@2	MSDialog MAP	MSDialog Recall@5	MSDialog Recall@1	MSDialog Recall@2
DMN-PRF	Only M1	0.7599	0.9294	0.6385	0.7761	0.5632	0.8509	0.3654	0.5579
	Only M2	0.7253	0.9271	0.5836	0.7440	0.4996	0.8584	0.2595	0.5021
	Inter-Dot (TB5)	**0.7719**	**0.9343**	**0.6552**	**0.7893**	**0.6792**	**0.9356**	**0.5021**	**0.7122**
	Inter-Cosine	0.7507	0.9260	0.6248	0.7675	0.6729	0.9356	0.4944	0.7027
	Inter-Bilinear	0.7228	0.9199	0.5829	0.7401	0.4923	0.8421	0.2647	0.4744
DMN-KD	Only M1	0.7449	0.9247	0.6167	0.7612	0.5776	0.8673	0.3805	0.5779
	Only M2	0.7052	0.9203	0.5538	0.7260	0.5100	0.8613	0.2794	0.5011
	Only M3	0.3887	0.6017	0.2015	0.3268	0.3699	0.6650	0.1585	0.2957
	M1+M2 (DMN)	0.7363	0.9196	0.6056	0.7509	0.6415	0.9155	0.4521	0.6673
	M1+M3	0.7442	0.9251	0.6149	0.7612	0.6134	0.8860	0.4224	0.6266
	M2+M3	0.7077	0.9198	0.5586	0.7263	0.5141	0.8659	0.2885	0.5069
	Inter-Dot (TB5)	**0.7655**	**0.9351**	**0.6443**	**0.7841**	0.6728	**0.9304**	0.4908	0.7089
	Inter-Cosine	0.7156	0.9121	0.5770	0.7268	**0.6916**	0.9249	**0.5241**	**0.7249**
	Inter-Bilinear	0.7061	0.9135	0.5590	0.7225	0.4936	0.8224	0.2679	0.4814

Useful Conclusions:

1. Dot product is the best.
2. PRF > KD

Learning to Paraphrase for Question Answering

Li Dong[†] and **Jonathan Mallinson**[†] and **Siva Reddy**[‡] and **Mirella Lapata**[†]
[†] ILCC, School of Informatics, University of Edinburgh
[‡] Computer Science Department, Stanford University
li.dong@ed.ac.uk, J.Mallinson@ed.ac.uk
sivar@stanford.edu, mlap@inf.ed.ac.uk

[2017-EMNLP]

Motivation

- Question answering (QA) is challenging due to the many different ways natural language expresses the same information need. As a result, small variations in semantically equivalent questions, may yield different answers.

"*who created microsoft*" vs "*who started microsoft*"

Incorporating external paraphrase knowledge

- Previous works are not generalized enough or trapped by low-quality paraphrases

Adaption

- Given question *q* and answer *a, estimate:*

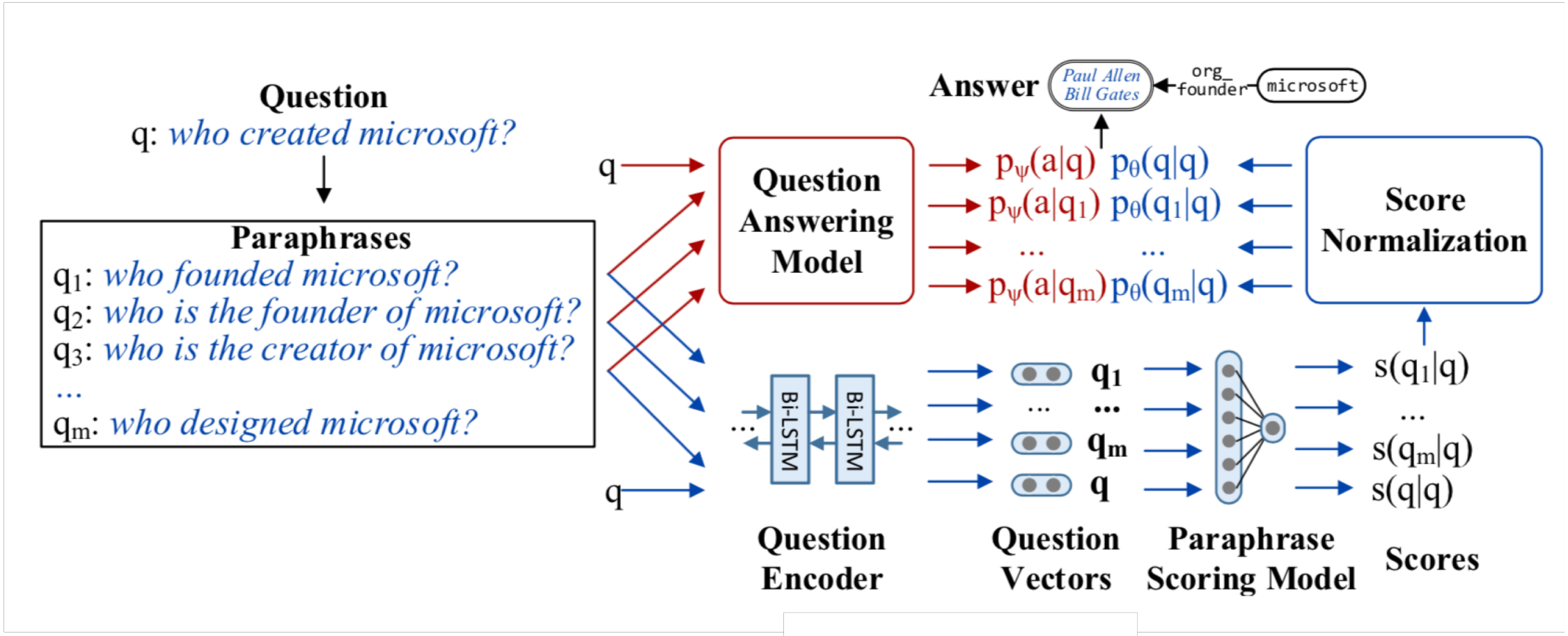
$$p(a|q)$$

$$p(a|q) = \sum_{q' \in H_q \cup \{q\}} \underbrace{p_\psi(a|q')}_{\text{QA Model}} \underbrace{p_\theta(q'|q)}_{\text{Paraphrase Model}}$$

Set of Paraphrases for Question *q*

Neural Scoring Model

- predicts the quality of the generated paraphrases
- learns to assign higher weights to those which are more likely to yield correct answers.



1. Encoding with Bi-LSTM hidden vectors

$$\mathbf{q} = \left[\overrightarrow{\mathbf{h}}_{|q|}, \overleftarrow{\mathbf{h}}_{1}\right] \qquad \mathbf{q} \in \mathbb{R}^{2n}$$

2. Scoring

$$s\left(q'|q\right) = \mathbf{w}_s \cdot \left[\mathbf{q}, \mathbf{q}', \mathbf{q} \odot \mathbf{q}'\right] + b_s$$

3. Normalization

$$p_\theta\left(q'|q\right) = \frac{\exp\{s\left(q'|q\right)\}}{\sum_{r \in H_q \cup \{q\}} \exp\{s\left(r|q\right)\}}$$

$$q' \in H_q \cup \{q\}$$

How to generate paraphrase?

- *PPDB-based Generation*:
  - replacing words and phrases (e.g. “*car*” with “*vehicle*”, “*manufacturer*” with “*producer*”)
- *NMT-based Generation:*
  - translating an English string into a foreign language and then back-translating it into English.
- *Rule-Based Generation:*
  - Fader et al., 2013[1]

The framework does not rely on specific paraphrase models.

[1] Anthony Fader, Luke Zettlemoyer, and Oren Etzioni. 2013. Paraphrase-driven learning for open question answering. In Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 1608–1618, Sofia, Bulgaria. Association for Computational Linguistics.

Results on WikiQA

Method	MAP	MRR
BIGRAMCNN (Yu et al., 2014)	0.6190	0.6281
BIGRAMCNN+CNT (Yu et al., 2014)	0.6520	0.6652
PARAVEC (Le and Mikolov, 2014)	0.5110	0.5160
PARAVEC+CNT (Le and Mikolov, 2014)	0.5976	0.6058
LSTM (Miao et al., 2016)	0.6552	0.6747
LSTM+CNT (Miao et al., 2016)	0.6820	0.6988
NASM (Miao et al., 2016)	0.6705	0.6914
NASM+CNT (Miao et al., 2016)	0.6886	0.7069
KVMEMNET+CNT (Miller et al., 2016)	**0.7069**	**0.7265**
BILSTM (baseline)	0.6456	0.6608
AVGPARA	0.6587	0.6753
SEPPARA	0.6613	0.6765
DATAAUGMENT	0.6578	0.6736
PARA4QA	**0.6759**	**0.6918**
−NMT	0.6528	0.6680
−PPDB	0.6613	0.6767
−RULE	0.6553	0.6756
BILSTM+CNT (baseline)	0.6722	0.6877
PARA4QA+CNT	**0.6978**	**0.7131**

Table 5: Model performance on WIKIQA. +CNT: word matching features introduced in Yang et al. (2015). The base QA model is BILSTM. Best results in each group are shown in **bold**.

Examples

Examples	$p_\theta\,(q'|q)$
(music.concert_performance.performance_role)	
what sort of part do queen play in concert	0.0659
what role do queen play in concert	0.0847
what be the role play by the queen in concert	0.0687
what role do queen play during concert	0.0670
what part do queen play in concert	0.0664
which role do queen play in concert concert	0.0652
(sports.sports_team_roster.team)	
what team do shaq play 4	0.2687
what team do shaq play for	0.2783
which team do shaq play with	0.0671
which team do shaq play out	0.0655
which team have you play shaq	0.0650
what team have we play shaq	0.0497

Table 6: Questions and their top-five paraphrases with probabilities learned by the model. The Freebase relations used to query the correct answers are shown in brackets. The original question is underlined. Questions with incorrect predictions are in *red*.

Story Ending Generation with Incremental Encoding and Commonsense Knowledge

Jian Guan[2*] **, Yansen Wang**[1*]**, Minlie Huang**[1†]
[1]Dept. of Computer Science & Technology, Tsinghua University, Beijing 100084, China
[1]Institute for Artificial IntelligenceTsinghua University (THUAI), China
[1]Beijing National Research Center for Information Science and Technology, China
[2]Dept. of Physics, Tsinghua University, Beijing 100084, China
guanj15@mails.tsinghua.edu.cn;ys-wang15@mails.tsinghua.edu.cn;
aihuang@tsinghua.edu.cn

[2019-AAAI]

Motivation

To consider

- representing the context clues which contain key information for planning a reasonable ending.
- using implicit knowledge (e.g., commonsense knowledge) to facilitate understanding of the story and better predict what will happen next.

Because, story ending generation requires more to deal with the logic and causality information that may span multiple sentences in a story context.

Examples

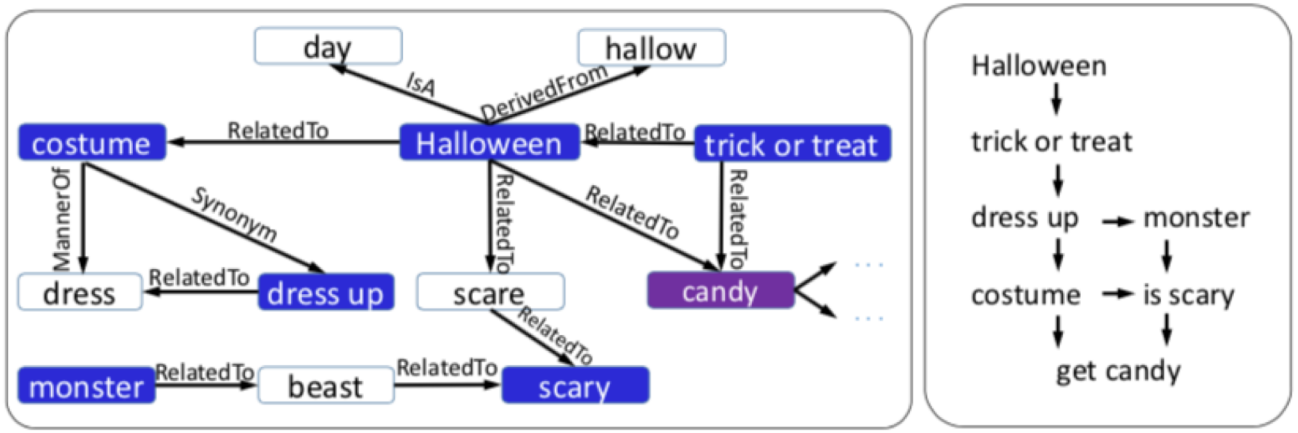
Halloween

Implicit Knowledge

Candy

Today is **Halloween** .
Jack is so excited to go **trick or treating** tonight .
He is going to **dress up** like a **monster** .
The **costume** is real **scary** .

He hopes to get a lot of **candy** .



Context Clue

Figure 1: A story example. Words in blue/purple are events and entities. The bottom-left graph is retrieved from ConceptNet and the bottom-right graph represents how events and entities form the context clue.

Formulation

Given

$$X = \{X_1, X_2, \cdots, X_K\}^3$$

$$X_i = x_1^{(i)} x_2^{(i)} \cdots x_{l_i}^{(i)}$$

To generate

$$Y = y_1 y_2 ... y_l \quad \Rightarrow \quad Y^* = \underset{Y}{argmax}\, \mathcal{P}(Y|X).$$

Model Overview

- Use seq2seq + attention:

$$\mathbf{h}_t = \mathbf{LSTM}(\mathbf{h}_{t-1}, e(x_t)), \quad \mathbf{s}_t = \mathbf{LSTM}(\mathbf{s}_{t-1}, e(y_{t-1}), \mathbf{c}_{t-1}),$$

Incremental Encoding Scheme

- To better encode context clue:

$$\mathbf{h}_j^{(i)} = \mathbf{LSTM}(\mathbf{h}_{j-1}^{(i)}, e(x_j^{(i)}), \mathbf{c}_{\mathbf{l}j}^{(i)}),\ i \geq 2.$$

Last Sentence

$$\mathbf{s}_t = \mathbf{LSTM}(\mathbf{s}_{t-1}, e(y_{t-1}), \mathbf{c}_{\mathbf{l}t}),$$

- Multi-Source Attention (MSA):

$$\mathbf{c}_{\mathbf{l}j}^{(i)} = \mathbf{W}_\mathbf{l}([\mathbf{c}_{\mathbf{h}j}^{(i)}; \mathbf{c}_{\mathbf{x}j}^{(i)}]) + \mathbf{b}_\mathbf{l},$$

- Impose supervision on encoder and decoder (better!):

$$\Phi = \Phi_{en} + \Phi_{de}$$

$$\Phi_{en} = \sum_{i=2}^{K}\sum_{j=1}^{l_i} -\log \mathcal{P}(x_j^{(i)} = \tilde{x}_j^{(i)} | x_{<j}^{(i)}, X_{<i}),$$

$$\Phi_{de} = \sum_{t} -\log \mathcal{P}(y_t = \tilde{y}_t | y_{<t}, X),$$

$$\mathcal{P}(y_t | y_{<t}, X) = \mathbf{softmax}(\mathbf{W}_0 \mathbf{h}_j^{(i)} + \mathbf{b}_0),$$

Multi-Source Attention (MSA)

$$\mathbf{c}_{\mathbf{l}j}^{(i)} = \mathbf{W}_{\mathbf{l}}([\mathbf{c}_{\mathbf{h}j}^{(i)}; \mathbf{c}_{\mathbf{x}j}^{(i)}]) + \mathbf{b}_{\mathbf{l}},$$

state context vector

$$\mathbf{c}_{\mathbf{h}j}^{(i)} = \sum_{k=1}^{l_{i-1}} \alpha_{h_k,j}^{(i)} \mathbf{h}_k^{(i-1)},$$

$$\alpha_{h_k,j}^{(i)} = \frac{e^{\beta_{h_k,j}^{(i)}}}{\sum_{m=1}^{l_{i-1}} e^{\beta_{h_m,j}^{(i)}}},$$

$$\beta_{h_k,j}^{(i)} = \mathbf{h}_{j-1}^{(i)\mathrm{T}} \mathbf{W}_{\mathbf{s}} \mathbf{h}_k^{(i-1)},$$

knowledge context vector

$$\mathbf{c}_{\mathbf{x}j}^{(i)} = \sum_{k=1}^{l_{i-1}} \alpha_{x_k,j}^{(i)} \mathbf{g}(x_k^{(i-1)}),$$

?

$$\alpha_{x_k,j}^{(i)} = \frac{e^{\beta_{x_k,j}^{(i)}}}{\sum_{m=1}^{l_{i-1}} e^{\beta_{x_m,j}^{(i)}}},$$

$$\beta_{x_k,j}^{(i)} = \mathbf{h}_{j-1}^{(i)\mathrm{T}} \mathbf{W}_{\mathbf{k}} \mathbf{g}(x_k^{(i-1)}),$$

from

ConceptNet
(Speer et al., 2012)

$$R = (h, r, t)$$

Knowledge Graph Representation

1. *Graph attention*
   - *Obtain* $\mathbf{G}(x) = \{R_1, R_2, \cdots, R_{N_x}\}$ (where each triple R_i has the same head concept x)
   - *Use G(x) to calculate g(x):*

$$\mathbf{g}(x) = \sum_{i=1}^{N_x} \alpha_{R_i} [\mathbf{h}_i; \mathbf{t}_i],$$

$$\alpha_{R_i} = \frac{e^{\beta_{R_i}}}{\sum_{j=1}^{N_x} e^{\beta_{R_j}}},$$

$$\beta_{R_i} = (\mathbf{W_r} \mathbf{r}_i)^{\mathrm{T}} tanh(\mathbf{W_h} \mathbf{h}_i + \mathbf{W_t} \mathbf{t}_i),$$

$$\mathbf{h}_i = \boldsymbol{e}(h_i), \mathbf{t}_i = \boldsymbol{e}(t_i)$$

Knowledge Graph Representation

2. Contextual attention

$$\mathbf{g}(x) = \sum_{i=1}^{N_x} \alpha_{R_i} \mathbf{M}_{R_i},$$

$$\mathbf{M}_{R_i} = BiGRU(\mathbf{h}_i, \mathbf{r}_i, \mathbf{t}_i),$$

$$\alpha_{R_i} = \frac{e^{\beta_{R_i}}}{\sum_{j=1}^{N_x} e^{\beta_{R_j}}},$$

$$\beta_{R_i} = \mathbf{h}_{(x)}^{\mathrm{T}} \mathbf{W_c} \mathbf{M}_{R_i},$$

Memory of *i-th* Triplet

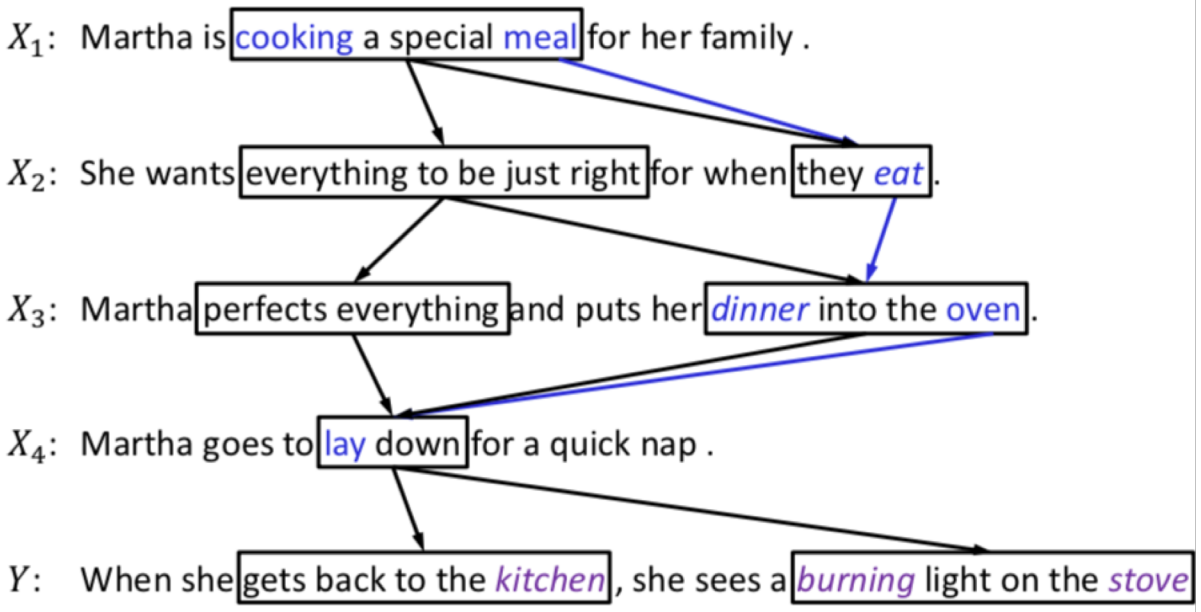
Evaluation Results

Model	PPL	BLEU-1	BLEU-2	Gram.	Logic.
Seq2Seq	18.97	0.1864	0.0410	1.74	0.70
HLSTM	17.26	0.2459	0.0771	1.57	0.84
HLSTM+Copy	19.93	0.2469	0.0783	1.66	0.90
HLSTM+MSA(GA)	15.75	0.2588	0.0809	1.70	1.06
HLSTM+MSA(CA)	12.53	0.2514	0.0825	1.72	1.02
IE (ours)	11.04	0.2514	0.0813	**1.84**	1.10
IE+MSA(GA) (ours)	9.72	0.2566	0.0854	1.68	**1.26**
IE+MSA(CA) (ours)	**8.79**	**0.2682**	**0.0936**	1.66	1.24

Manual Evaluation

$$PP(W) = P(w_1 w_2 \dots w_N)^{-\frac{1}{N}} = \sqrt[N]{\frac{1}{P(w_1 w_2 \dots w_N}}$$

Case Study



Entity	commonsense knowledge
cook	(cook, AtLocation, *kitchen*)
	(cook, HasLastSubevent, *eat*)
meal	(meal, AtLocation, *dinner*)
	(meal, RelatedTo, *eat*)
eat	(eat, AtLocation, *dinner*)
oven	(oven, AtLocation, *stove*)
	(oven, RelatedTo, *kitchen*)
	(oven, UsedFor, *burn*)

Figure 3: An example illustrating how incremental encoding builds connections between context clues.

Now I'm trying to use these external knowledge:

I. Relevant QA Text → key-value
II. Paraphrase (e.g. PPDB) → symmetry pair
III. ConceptNet → graph

Thanks!