Goal-oriented search for dialogue and question answering

Jian-Yun Nie

University of Montreal

nie@iro.umontreal.ca

http://rali.iro.umontreal.ca/nie-site/jian-yun-nie-en/

Context

- Search is used as a basic tool in many applications and tasks
  - Retrieval-based chatbot
  - Question answering
  - Social media analysis
  - …
- Usually, simple IR methods to find relevant documents
  - Containing query words
- Are the search results the most appropriate for the task?

Search in the tasks

- Retrieval-based chatbot
  - Use the current utterance to retrieve relevant responses from the repository
- Question answering
  - Use the question to retrieve relevant documents/passages
- Social media analysis
  - Use queries (keywords) to retrieve tweets

- All these searches focus on *relevance: containing the words*
  - *Does a retrieved response fit the context and goal?*
  - *Does a retrieved passage contain an answer?*
  - *Are the tweets about the specific event (person/product, …) and the intended analysis?*

Proposal: considering the goal in search

- Retrieval-based dialogue
  - Responses should fit the context and the goal
- Question answering
  - Retrieved passages should contain potential answers
- Query expansion is not enough
  - Query and Context/Goal should be used in different ways

This talk

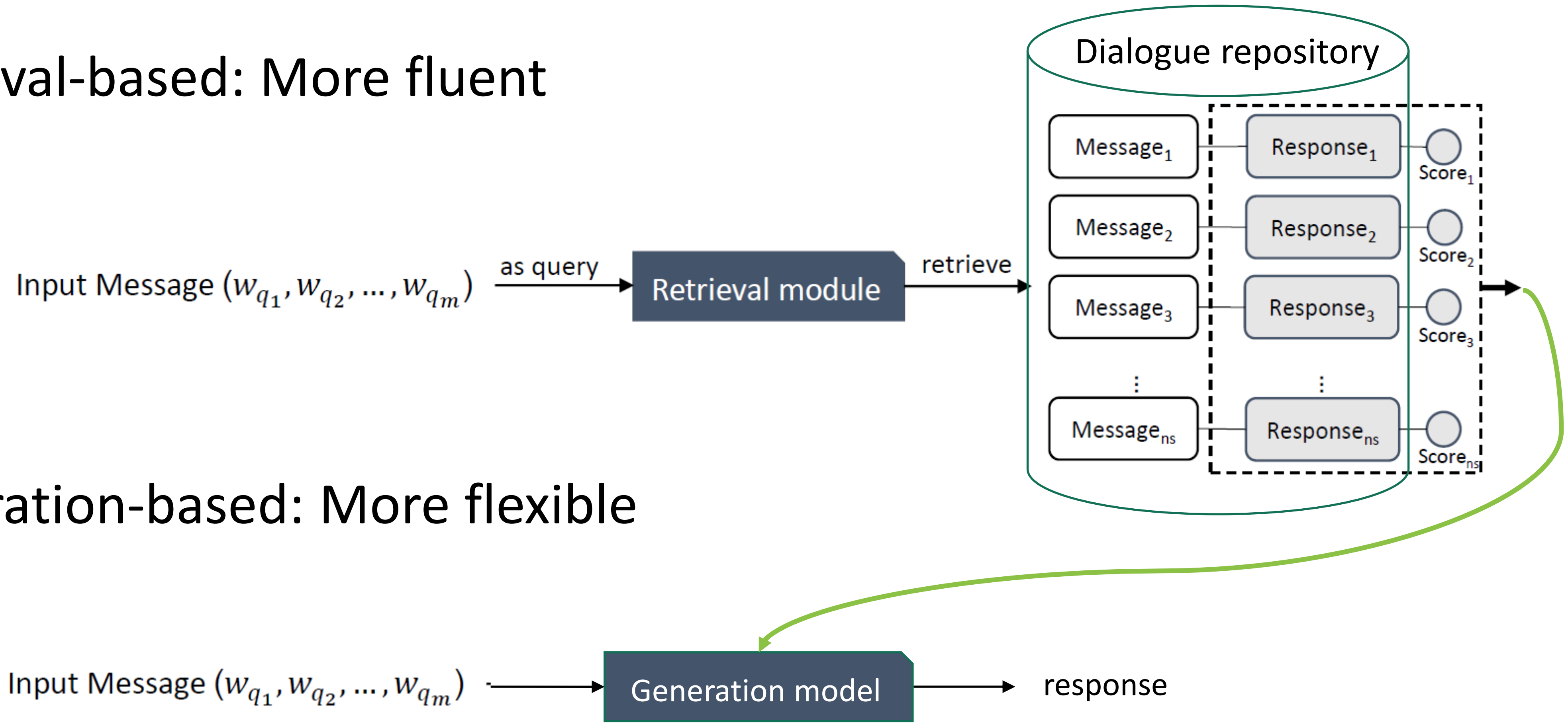
- Dialogue with users
  - Response generation enhanced by retrieved responses
  - Incorporating the goal (proactive dialogue)
- Question answering
  - Answer-oriented passage reranking

Dialogue

- Chitchat: generate / retrieve a reasonable response
- Task-oriented conversation
- Dialogue with a goal, based on knowledge: Guide conversation in a direction
- Provide an answer (QA): finding an answer from documents / knowledge graph
- …

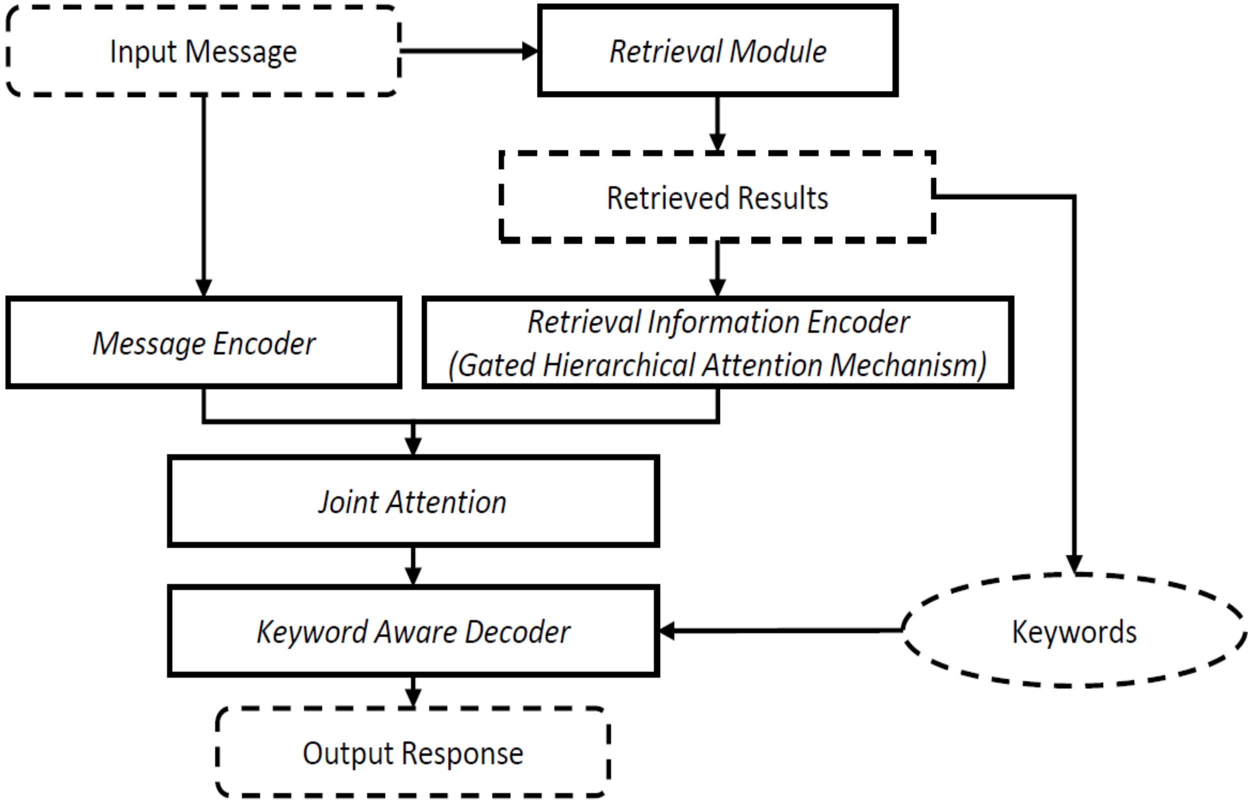
Retrieval-based vs. Generation-based

- Retrieval-based: More fluent
- Generation-based: More flexible



1. Retrieved candidates serve as context for response generation

- Issue with generation-based responses
  - Often bland: I don't know, I don't understand what you say, ..
- Issue with retrieval-based responses
  - Limited choices
- Expected advantages of a combined approach
  - Reuse part of the existing responses (know how to respond)
  - Generate new responses



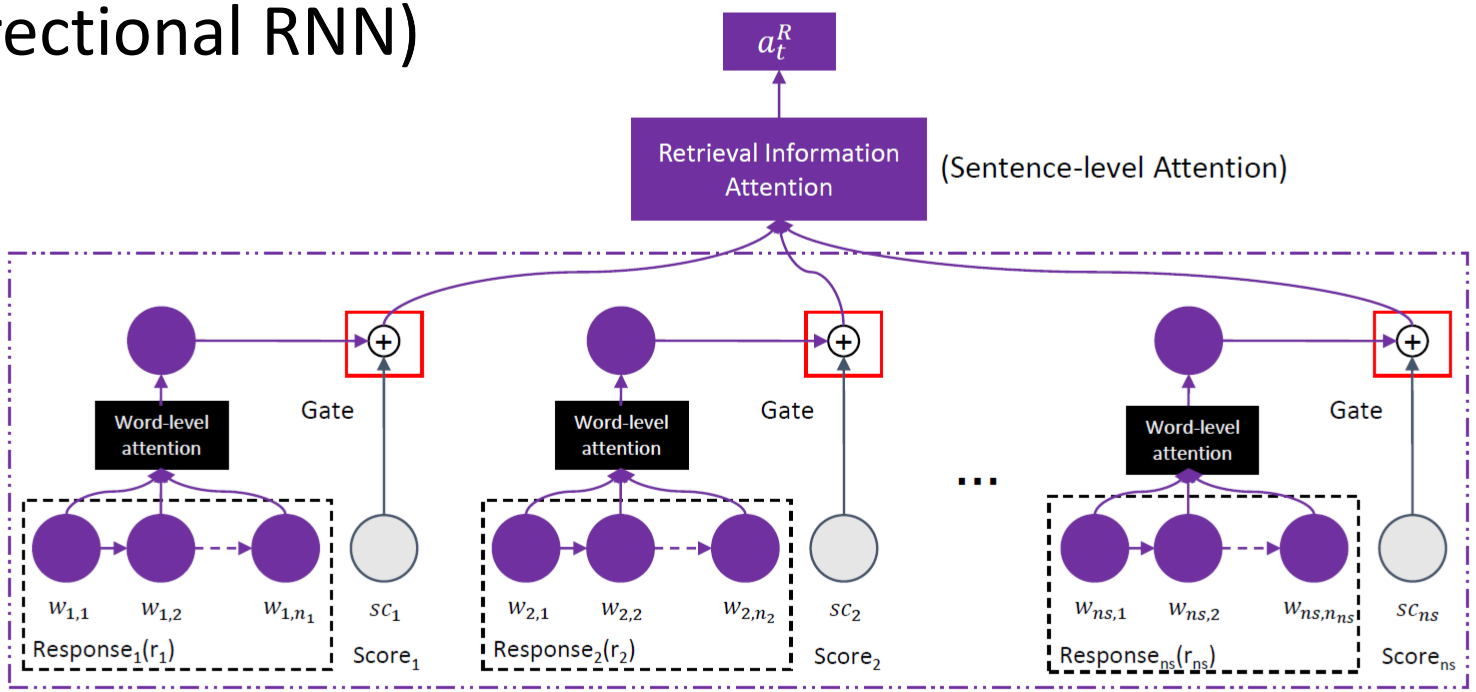
Yutao Zhu, Zhicheng Dou, Jian-Yun Nie, Ji-Rong Wen: ReBoost: a retrieval-boosted sequence-to-sequence model for neural response generation. Inf. Retr. J. 23(1): 27-48 (2020)

Some examples

Input Message	Gold Response	Retrieved Results
I'm **bored**, is **anyone** awake? Anyone want to **chat**? 无聊中，有没有没睡觉的？聊下？	*Let's go, I <u>haven't slept</u> yet.* *来来，我也还<u>没睡觉</u>*	I'm **bored**! Is there **anyone** who can **chat** with me? I will reply to anything. 无聊中！有陪聊的没？有来必应。 *It seems that many people are <u>not sleeping</u>.* *看来不少人都<u>没睡觉</u>哦*
A fresh grown **watermelon**, **have you ever seen one?** 刚结出来的西瓜，你见过吗？	*This is too tiny. Such a small watermelon. <u>Is it edible?</u>* *这未免太迷你了，好袖珍的西瓜，<u>能吃吗？</u>*	A blue **watermelon**, is it beautiful? **Have you ever seen one?** 蓝色的西瓜，美不美？你见过吗？ *Such a weird watermelon. <u>Is it edible?</u>* *好奇怪的西瓜，<u>这个能吃吗？</u>*
The **Beijing** today **is most suitable for sleeping** 今天的北京最适合睡觉	*<u>Sleeping</u> on rainy <u>days</u> is refreshing!* *下雨<u>天睡觉</u>爽！*	It's too humid in **Beijing** … This type of day **is most suitable for sleeping** 北京湿透了…这天最适合睡觉了 *Today's plan: <u>sleep</u> all <u>day</u>.* *今天的计划，全<u>天睡觉</u>。*

Response retrieval and encoding

- Retrieval with a traditional IR method (BM25)
- Response encoder (Bi-directional RNN)
  - Sentence encoding
  - Sentence aggregation



Response generation

- Input: concatenation of message and retrieved responses

$$\mathbf{a}_t = [\mathbf{a}_t^M; \mathbf{a}_t^R],$$

- Keyword-guided generation (keywords from responses by TF-IDF)

$$p_t = p_n$$
$$p_n = \mathrm{softmax}(\mathbf{W}_s \mathbf{s}_t + \mathbf{b}_s),$$

p_k: prob. of keyword selected from the responses
p_n: normal generation prob.

$$\mathbf{s}_t = \mathrm{GRU}(\mathbf{y}_{t-1}, [\mathbf{s}_{t-1}; \mathbf{a}_t]),$$

Some results

- On NTCIR-13 STC task (Chinese)
- Open Subtitles (English)

(a) Results on Weibo Dataset

	Distinct-1	Distinct-2	BLEU-1	BLEU-2	BLEU-3	BLEU-4
S2SA	.0107	.0499	7.25	2.77	1.46	0.93
NRM-hyb	.0142	.0699	11.66	4.69	2.70	1.90
MMI	.0132	.0683	12.70	4.66	2.52	1.69
TA-Seq2seq	.0133	.0671	12.30	4.59	2.52	1.85
ReBoost	**.0302**	**.2112**	**12.73**	**5.68**	**3.55**	**2.62**

(b) Results on OpenSubtitles Dataset

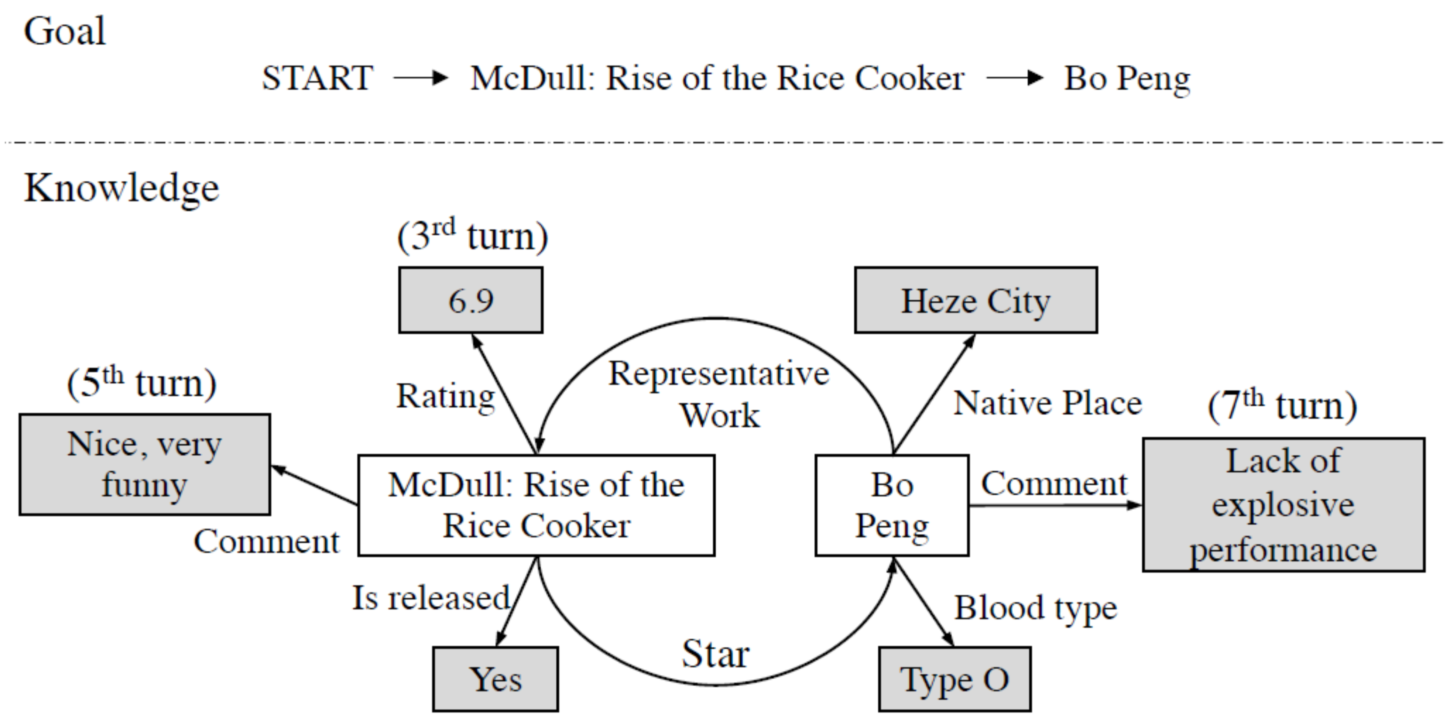
	Distinct-1	Distinct-2	BLEU-1	BLEU-2	BLEU-3	BLEU-4
S2SA	.0025	.0078	6.84	2.59	1.4	0.75
NRM-hyb	.0025	.0080	6.57	2.7	1.46	0.77
MMI	.0015	.0062	**8.83**	3.36	1.76	0.9
TA-Seq2seq	.0026	.0089	5.04	2.03	1.16	0.7
ReBoost	**.0027**	**.0090**	8.57	**3.93**	**2.21**	**1.49**

2. Proactive Dialogue – meet a conversation goal

- Chitchat: without specific purpose
- Task-oriented dialogue: specific to a task (patterns, slots, …)
- Can we design a general chatbot for some purpose?
  - To express *happy* emotion
  - To mention some entities (recommendation)
  - …

Knowledge-grounded proactive conversation

- Movie domain
- The chatbot should lead the conversation to discuss about some entities (movie, actor, …)
- Based on knowledge



Desired conversation

Goal
START ⟶ McDull: Rise of the Rice Cooker ⟶ Bo Peng

(1) Bot	Do you usually spend your weekend watching movies? (平时周末去看电影吗?)
(2) User	Of course, could you recommend good films for me? I can watch it on this weekend. (看呀，最近有好看的可以推荐一下，周末去看。)
(3) Bot	You can watch **Rise of the Rice Cooker**, which is rated 6.9. (麦兜·饭宝奇兵这部电影，大众评分在6.9呢，值得一看。)
(4) User	Sounds great, thanks. (好的，一定去看看。)
(5) Bot	It is very nice and funny. Do you know who is the star of this movie? (这部电影还不错，挺搞笑的。你知道饭麦兜·饭宝奇兵的主演是谁吗?)
(6) User	I don’t know. Who is that? (不知道，谁啊?)
(7) Bot	It’s **Bo Peng**. Do you know him? People said that he was a kind of lacking explosive Performance. (彭博，网友们评论说他缺少爆发力太温柔)
(8) User	I will know more about him later. (那我有时间去了解一下。)

Approach: integrating the goal and knowledge

- User's utterance (query): $\mathbf{e}^{u_i}$
- Goal: $\mathbf{e}^{g}$
- Select relevance piece of knowledge: $\mathbf{e}^{k_j}$
- Candidate response: $\mathbf{e}^{r}$

Matching features

- Word-level similarity:

$$\mathbf{m}_1^{u_i} = [\mathbf{e}^{u_i}\mathbf{A}_1\mathbf{e}^{r}; \cos(\mathbf{e}^{u_i}, \mathbf{e}^{r})],$$
$$\mathbf{m}_1^{k_j} = [\mathbf{e}^{k'_j}\mathbf{A}_1\mathbf{e}^{r}; \cos(\mathbf{e}^{k'_j}, \mathbf{e}^{r})],$$

- Sentence-level similarity: (LSTM)

$$\mathbf{m}_2^{u_i} = [\mathbf{u}_i\mathbf{A}_2\mathbf{r}; \cos(\mathbf{u}_i, \mathbf{r})],$$
$$\mathbf{m}_2^{k_j} = [\mathbf{k}_j\mathbf{A}_2\mathbf{r}; \cos(\mathbf{k}_j, \mathbf{r})],$$

CNN

$$\mathbf{v}^{u} = [\mathbf{v}^{u_1}, \cdots, \mathbf{v}^{u_L}]$$
$$\mathbf{v}^{k} = [\mathbf{v}^{k_1}, \cdots, \mathbf{v}^{k_M}]$$

$$\hat{y} = \big(s(c,r) + s(k,r) + s(g,r)\big)/3.$$

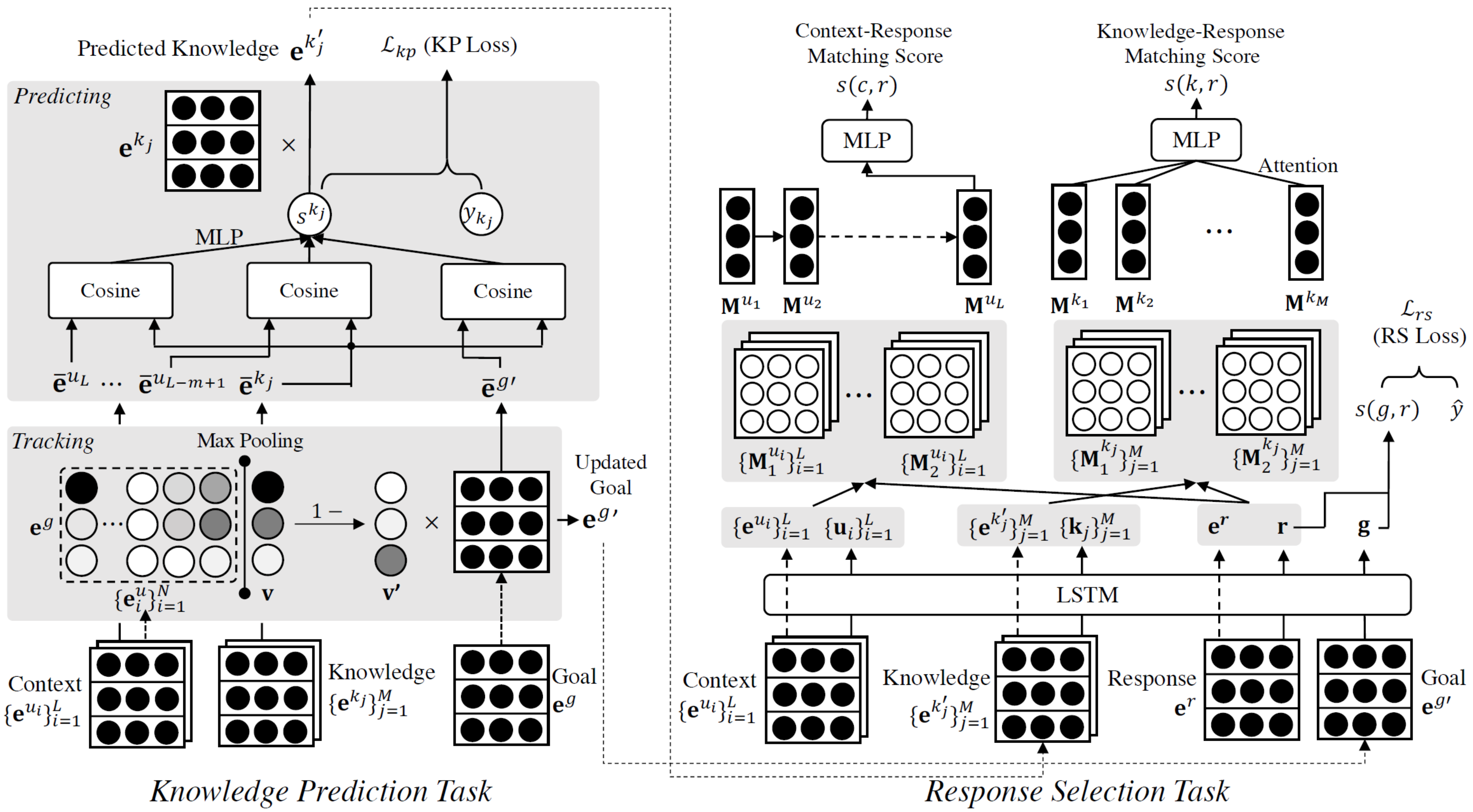
Some more details

- Selecting relevance pieces (triples) of knowledge, based on goal and utterance
  - Use a separate training to learn to use knowledge graph

$$\mathcal{L}_{kp} = -\frac{1}{|\mathcal{D}|} \sum \left(y_{k_j} \log s^{k_j} + (1 - y_{k_j}) \log(1 - s^{k_j}) \right)$$

- Goal tracker
  - Track what part of the goal has been met, and what remains

Architecture



Some results

	Hits@1	Hits@3	MRR	BLEU1	BLEU2	KLG. P	KLG. R	KLG. F1	KLG. Acc.	Goal Acc.
DuConv										
Ground-truth	-	-	-	1.00	1.00	38.24	9.20	14.83	100.00	100.00
DuRetrieval	50.12	75.68	63.13	0.47	0.32	30.11	7.24	11.68	53.64	58.90
KPN	**66.94**	**87.52**	**78.30**	**0.56**	**0.42**	**33.45**	**8.05**	**12.97**	**57.82**	**77.58**

- DuRetrieval: A naïve use of knowledge and goal (mixed)
- KPN: using knowledge and goal

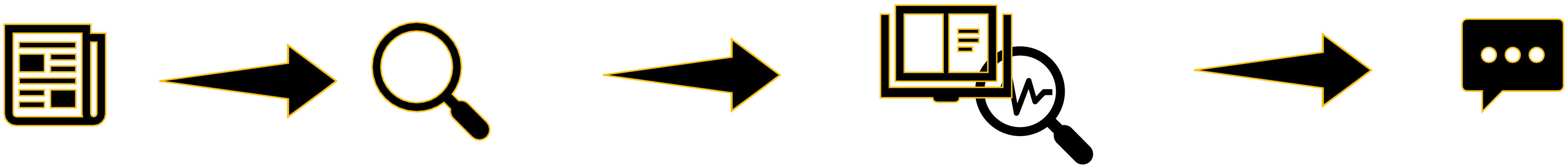
Observations

- Both goal and knowledge are useful for retrieval-based conversation
- Different ways to use knowledge and goal (different paths of matching)

- Matching for the purpose of retrieving a response that fits the goal, based on knowledge

3. Open-domain QA:
Answer-oriented passage reranking

- Open domain QA: find answer from many texts



Retriever: retrieve a set of texts (documents/ paragraphs/sentences) for a question

Reader: Machine reading comprehension (MRC) to find an answer from the selected documents

Key issue: Retriever and Reader disconnected

Illustration Example

	Question:			**What Russian emigre to the U.S. is credited with inventing the helicopter ?**
	Ground Truth:			**Igor**
		r	a	Passages
✗	P_1	1	0	Wright brothers: Orville and Wilbur Wright, were the two Americans who are credited with inventing and building the world's first successful airplane.
?	P_2	1	1	The fellow Russian emigre, ~~**Igor**~~ Ivanovich Sikorsky, was an American aviation pioneer in both helicopters and fixed-wing aircraft, who was credited with many other accomplishments...
	P_3	0	1	~~**Igor**~~ Ivanovich Sikorsky was an orthodox chiristian.
✓	P_4	1	1	His paternal grandfather, Leo Shoumatoff, was the bussiness manager of ~~**Igor**~~ Sikorsky's aircraft company, where **Igor** developed the first helicopter and the first passenger airplane.

- Ranking by retriever: P2, P4, P1, P3
- Some of the passages contain query words, but do not contain the answer (P1) or do not support the answer (P2)
- Idea: Select passages that are relevant and may contain the answer (lightweight reader)

Background - Problems

- Current trend: increase the model size
  - Pretrained LMs as KBs: GPT-3 (175B), Switch Transformer (1.6T)
  - A 11B T5 model matches the performance with DPR with 3 BERT-based models (Roberts etc., 2020)
  - Heavy machinery is required to run them.
- Building a large Neural retrieval model
  - Dense Passage Retrieval (DPR): also needs a large GPU server to run

Adding an answer-oriented passage selector

- Passage reranking: Relevance + Containing possible answer (lightweight reader)

Retriever: retrieve a set of texts (documents/ paragraphs/sentences) for a question

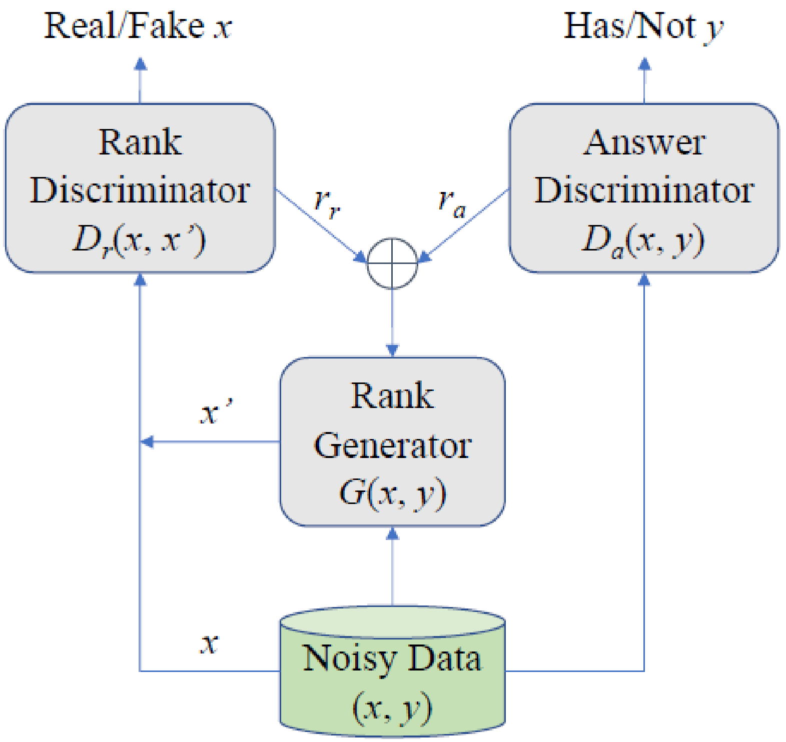
Ranker: select/rerank passages according to relevance + possible answer

Reader: Machine reading comprehension to find an answer from the selected documents

Training of Ranker with noisy data

- We are limited by the amount of training data (question-passage-answer)
- Available: Question-answer
- Assumption: a passage containing the answer is a good passage

Question:			**What Russian emigre to the U.S. is credited with inventing the helicopter ?**
Ground Truth:			**Igor**
	r	**a**	Passages
P_1	1	0	Wright brothers: Orville and Wilbur Wright, were the two Americans who are credited with inventing and building the world's first successful airplane.
P_2	1	1	The fellow Russian emigre, ~~**Igor**~~ Ivanovich Sikorsky, was an American aviation pioneer in both helicopters and fixed-wing aircraft, who was credited with many other accomplishments...
P_3	0	1	~~**Igor**~~ Ivanovich Sikorsky was an orthodox chiristian.
$\mathbf{P_4}$	1	1	His paternal grandfather, Leo Shoumatoff, was the bussiness manager of ~~**Igor**~~ Sikorsky's aircraft company, where **Igor** developed the first helicopter and the first passenger airplane.



GAN-based training

- Try to separate good and bad examples
- General GAN:
  - Generator – learns the distribution of true data
  - Discriminator – tries to separate true and fake data
- Extended GAN framework
  - Generator
  - 2 discriminators: relevant and contain answer?

Some formulas

- Overall objective

$$
\begin{aligned}
J = \min_{\theta} \max_{\phi,\xi} \sum_{n=1}^{N} \Big(& \mathbb{E}_{d \sim p_{true}(d|q_n,a)} [\log D_{\phi}^{r}(d|q_n)] + \\
& \mathbb{E}_{d \sim p_{\theta}(d|q_n,a)} [\log(1 - D_{\phi}^{r}(d|q_n))] - \\
& \lambda_1 \cdot \mathbb{E}_{d \sim p_{\theta}(d|q_n,a)} [\log D_{\xi}^{a}(d|q_n)] + \\
& \lambda_2 \cdot \mathbb{E}_{d \sim p_{true}(d|q_n,a)} [\log \frac{p_{true}(d|q_n, a)}{p_{\theta}(d|q_n, a)}] \Big)
\end{aligned}
$$

- Generator: $p_{\theta}(d|q_n, a)$
- Rank discriminator (relevance): D_{ϕ}^{r}
- Answer discriminator: D_{ξ}^{a}
- Regularizer: $\mathbb{E}_{d \sim p_{true}(d|q_n,a)} [\log \frac{p_{true}(d|q_n, a)}{p_{\theta}(d|q_n, a)}]$

Some more formulas for losses

- Rank discriminator:

$$\mathcal{L}_{D^r_\phi} = -\sum_{n=1}^{N}\Big(\mathbb{E}_{d\sim p_{true}}[\log(\sigma(f_\phi(d,q_n)))]+ \mathbb{E}_{d\sim p_{\theta^*}}[\log(1-\sigma(f_\phi(d,q_n)))]\Big)$$

- Answer discriminator:

$$\mathcal{L}_{D^a_\xi} = -\sum_{n=1}^{N}\Big(\sum_{d\in A^+}\log\sigma(f_\xi((d,q_n)))+ \sum_{d\in A^-}\log(1-\sigma(f_\xi(d,q_n)))\Big)$$

- Generator:

$$\mathcal{L}_{p_\theta} = \sum_{n=1}^{N}\Big(\mathbb{E}_{d\sim p_\theta}[\log(1-\sigma(f_{\phi^*}(d,q_i)))]- \lambda_1\,\mathbb{E}_{d\sim p_\theta}[\log\sigma(f_{\xi^*}(d,q_n))]- \lambda_2\,\mathbb{E}_{d\sim p_{true}}[\log p_\theta(d|q_n,a)]\Big)$$

Training

- Document and question encoding: BiLSTM + self-attention
- Score functions in discriminators: $f_\phi(d_i, q) = p(d_i|q) = \text{softmax}(\max_j(\hat{d}_i^j W q)),$
- Score by generator: $f_\theta(d_i, q) = p_\theta(a|q, d_i) = \max_{j,k} p_s^j(a|q, d_i) p_e^k(a|q, d_i)$
- REINFORCE algorithm for training

Retriever and Reader

- Retriever: BM25
- Reader: A more sophisticated reader based on 12-layer BERT

$$P(s, e, i) = P(d_i) \cdot P(s|d_i) \cdot P(e|d_i)$$

$$P(s|d_i) = \text{softmax}(d_i w_{start})_s$$

$$P(e|d_i) = \text{softmax}(d_i w_{end})_t$$

$$P(d_i) = \text{softmax}(\hat{D}^T w_{doc})_i$$

Some experimental results

- Test collections

Dataset	#Train	#Dev	#Test	#Psgs/Que
Quasar-T	37,012	3,000	3,000	100
SearchQA	99,811	13,893	27,247	~49.6
TriviaQA	87,291	11,274	10,790	100
CuratedTREC	1,353	133	694	Wikipedia (50)
Nat.Question	79,168	8,757	3,610	Wikipedia (50)

- Reranking (part)

	Quasar-T			SearchQA		
	BM25	DSQA	Ours	BM25	DSQA	Ours
Hits@1	*6.3*	*27.7*	**35.2**	*13.7*	*59.9*	**63.9**
Hits@3	*10.9*	*36.8*	**52.0**	*24.1*	*69.8*	**83.0**
Hits@5	*15.2*	*42.6*	**59.5**	*32.7*	*75.5*	**88.8**
Hits@20	-	-	**72.3**	-	-	**97.5**
Hits@50	-	-	**74.8**	-	-	**99.8**

Final answer

	Quasar-T	SearchQA	Cur.Trec	Trivia	NQ_{sub}
BM25	41.6	57.9	*21.3*	*47.1*	26.7
R^3	*35.3*	*49.0*	*28.4*	*47.3*	-
DSQA	*42.2*	*58.8*	*29.1*	*48.7*	-
DPR	-	-	*28.0*	*57.0*	27.4
Ours	**45.5**	**61.2**	**29.3**	**60.7**	**29.5**

Example

Question: **What Russian emigre to the U.S. is credited with inventing the helicopter ?**
Ground Truth: **Igor**

	r	**a**	Passages	DPR Score	Our Score
P_1	1	0	Wright brothers: Orville and Wilbur Wright, were the two Americans who are credited with inventing and building the world's first successful airplane.	71.7	242.9
P_2	1	1	The fellow Russian emigre, ~~***Igor***~~ Ivanovich Sikorsky, was an American aviation pioneer in both helicopters and fixed-wing aircraft, who was credited with many other accomplishments...	**75.6**	286.8
P_3	0	1	~~***Igor***~~ Ivanovich Sikorsky was an orthodox chiristian.	57.7	178.1
$\mathbf{P_4}$	1	1	His paternal grandfather, Leo Shoumatoff, was the bussiness manager of ~~***Igor***~~ Sikorsky's aircraft company, where **Igor** developed the first helicopter and the first passenger airplane.	74.4	**292.3**

Conclusion on answer-oriented passage reranking

- Reranking based on possible answer in the passage is useful
- GAN can better leverage noisy training data
- Lightweight ranker: a fraction of retrieval and machine reading time
  - Retriever: 3.3 ms
  - Ranker: 0.5 ms
  - Reader: 57.3 ms

Final notes

- Tailor search for different purposes
  - Design specific ways to enhance matching
- Towards a more general framework of goal-oriented search
  - A large body of work in the literature about the principles
  - Need to find ways to implement them
  - → Neural nets offer a flexible framework for the implementation

- Additional issues:
  - Training and test data with user interactions
  - A new evaluation methodology

Thank you !