Paper Reading

Jun Gao

June 26, 2018

Tencent AI Lab

Neural Generative Question Answering [IJCAI2016]

Introduction

This paper presents an end-to-end neural network model, named Neural Generative Question Answering (GENQA), that can generate answers to simple factoid questions, based on the facts in a knowledge-base.

- The model is built on the encoder-decoder framework for sequence-to-sequence learning, while equipped with the ability to enquire a knowledge-base
- Its decoder can switch between generating a common word and outputting a term) retrieved from knowledge-base with a certain probability.
- The model is trained on a dataset composed of real world question-answer pairs associated with triples in the knowledge-base.

The GENQA Model

The GENQA model consists of **Interpreter**, **Enquirer**, **Answerer**, and an external knowledgebase. **Answerer** further consists of **Attention Model** and **Generator**.

- **Interpreter** transforms the natural language question Q into a representation $\boldsymbol{H}_Q$ and saves it in the short-term memory.
- **Enquirer** takes $\boldsymbol{H}_Q$ as input to interact with the knowledge-base in the long-term memory, retrieves relevant facts (triples) from the knowledge-base, and summarizes the result in a vector $\boldsymbol{r}_Q$.
- The **Answerer** feeds on the question representation $\boldsymbol{r}_Q$ as well as the vector $\boldsymbol{r}_Q$ and generates an answer with Generator.

The GENQA Model

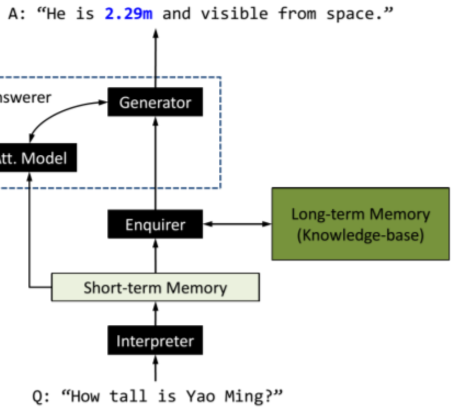

Figure 1: System diagram of GENQA.

Interpreter

Given the question represented as word sequence $Q = (x_1, ...x_{T_Q})$, Interpreter encodes it to an array of vector representations.

- In our implementation, we adopt a bi-directional recurrent neural network(GRU).
- By concatenating the hidden states (denoted as $(\boldsymbol{h}_1, ..., \boldsymbol{h}_{T_Q})$), the embeddings of words ((denoted as $(\boldsymbol{x}_1, ..., \boldsymbol{x}_{T_Q})$) , and the one-hot representations of words, we obtain an array of vectors $\boldsymbol{H}_Q = (\tilde{\boldsymbol{h}}_1, ..., \tilde{\boldsymbol{h}}_{T_Q})$, where $\tilde{\boldsymbol{h}}_t = [\boldsymbol{h}_t; \boldsymbol{x}_t; x_t]$.
- This array of vectors is saved in the short-term memory, allowing for further processing by Enquirer and Answerer.

Interpreter

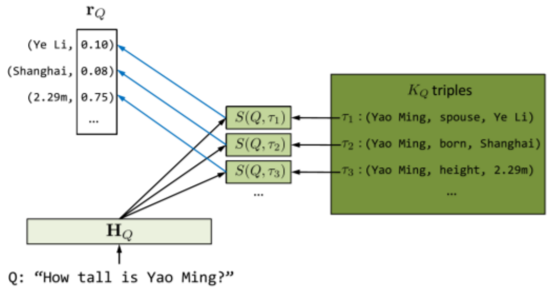

Figure 2: Enquirer of GENQA.

Enquirer

- Enquirer first performs term-level matching to retrieve a list of relevant candidate triples, denoted as $\tau_Q = \{\tau_k\}_{k=1}^{k_Q}$. k_Q is the number of candidate triples.
- After obtaining τ_Q, Enquirer calculates the relevance (matching) scores between the question and the K_Q triples. The k^{th} element of $\boldsymbol{r}_Q$ Q is defined as the probability

$$r_{Q_k} = \frac{e^{S(Q,\tau_k)}}{\sum_{k'=1}^{K_Q} e^{S(Q,\tau_{k'})}}$$

- where $S(Q, \tau_k)$ denotes the matching score between question Q and triple τ_k.The probability in $\boldsymbol{r}_Q$ will be further taken into the probabilistic model in Answerer for generating an answer.

Enquirer

In this work, we provide two implementations for Enquirer to calculate the matching scores between question and triples.

- Bilinear Model: simply takes the average of the word embedding vectors in $\boldsymbol{H}_Q$ as the representation of the question (with the result denoted as $\bar{\boldsymbol{x}}_Q$).

$$\bar{S}(Q, \tau) = \bar{\boldsymbol{x}}_Q^T \boldsymbol{M} \boldsymbol{u}_\tau$$

  where M is a matrix parameterizing the matching between the question and the triple.
- CNN-based Matching Model: the question is fed to a convolutional layer followed by a max-pooling layer, and summarized as a fixed-length vector $\hat{\boldsymbol{h}}_Q$.

$$\bar{S}(Q, \tau) = f_{MLP}([\hat{\boldsymbol{h}}_Q; \boldsymbol{u}_\tau])$$

Answerer

- Answerer uses an RNN to generate an answer based on the information of question saved in the short-term memory (represented as $\boldsymbol{H}_Q$) and the relevant facts retrieved from the long-term memory (indexed by $\boldsymbol{r}_Q$).
- In generating the t^{th} word y_t t in the answer, the probability is given by the following mixture model

$$p(y_t|y_{t-1}, s_t, \boldsymbol{H}_Q, \boldsymbol{r}_Q; \theta) =$$
$$p(z_t = 0|s_t; \theta)p(y_t|y_{t-1}, s_t, \boldsymbol{H}_Q, z_t = 0; \theta)+$$
$$p(z_t = 1|s_t; \theta)p(y_t|\boldsymbol{r}_Q, z_t = 1; \theta)$$

  which sums the contributions from the language part and the knowledge part, with the coefficient $p(z_t|s_t; \theta)$ being realized by a logistic regression model with s_t as input.

Answerer

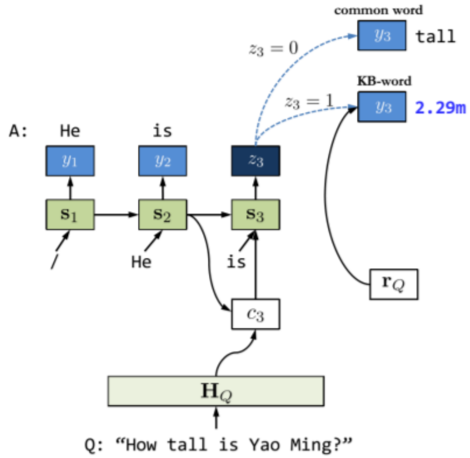

Figure 3: Answerer of GENQA.

Results

Table 4: Test accuracies

Models	Test
Retrieval-based QA	36%
NRM[13]	19%
Embedding-based QA [7]	45%
GENQA	47%
GENQA$_{\text{CNN}}$	**52%**

Examples

Figure 4: Examples of the generated answers by GENQA.

ID	Questions	Generated Answers
1	三国演义谁写的? Who wrote the Romance of the Three Kingdoms?	明朝的罗贯中。 Luo Guanzhong in Ming dynasty.
2	项羽是哪国人? Which state does Xiang Yu belong to?	楚国的皇帝。 King of the Chu State.
3	还珠格格多少集? How many episodes does My Fair Princess have?	一共24集。 24 episodes in total.
4	泰戈尔是哪国作家? Which country is the writer Tagore from?	泰戈尔是印度著名的作家 。 Tagore is a famous Indian writer.
5	孙燕姿今年多大啊? How old is Stefanie Sun this year?	32岁，他出生于1978年7月23日。 Thirty-two, he was born on July 23, 1978.
6	《格列佛游记》的作者是谁? Who is the author of Gulliver's Travels?	法国作家斯威夫特。 The French writer, Swift.
7	怪物史莱克4什么时候上映呀? When will Shrek Forever After be released?	上映日期：梦工厂电影发行公司。 Release date: DreamWorks Pictures.
8	凤凰卫视是哪里的电视台? Which place is phoenix television located at?	总部在刘长乐。 The headquarters is at Liu Changle.

Conclusion

The model is built on the encoder-decoder framework for sequence-to-sequence learning, while equipped with the ability to query a knowledge-base.

A Knowledge-Grounded Neural Conversation Model [AAAI2018]

Introduction

This paper presents a novel, fully data-driven, and knowledge-grounded neural conversation model aimed at producing more contentful responses.

- It offers a framework that generalizes the SEQ2SEQ approach of most previous neural conversation models, as it naturally combines conversational and non-conversational data via multi-task learning.

Grounded Response Generation

In order to infuse the response with factual information relevant to the conversational context, we propose a knowledge-grounded model architecture.

- First, we have available a large collection of world facts, which is a large collection of raw text entries indexed by named entities as keys.
- Then, given a conversational history or source sequence S, we identify the focus in S,which is the text span based on which we form a query to link to the facts.
- Finally, both conversation history and relevant facts are fed into a neural architecture that features distinct encoders for conversation history and facts.

Grounded Response Generation

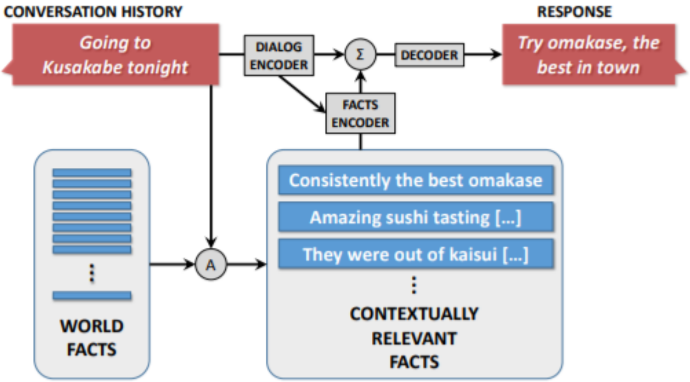

Figure 3: Knowledge-grounded model architecture.

Dialog Encoder and Decoder

- The dialog encoder and response decoder form together a sequence-to-sequence (SEQ2SEQ model)
- This part of our model is almost identical to prior conversational SEQ2SEQ models, except that we use gated recurrent units (GRU) instead of LSTM cells.

Facts Encoder

Given an input sentence $S = \{s_1, s_2, ..., s_n\}$,and a fact set $F = \{f_1, f_2, ..., f_k\}$ The RNN encoder reads the input string word by word and updates its hidden state.

- u is the summary of the input sentence and r_i is the bag of words representation of f_i. The hidden state of the RNN is initialized with $\hat{u}$ to predict the response sentence R word by word.

$$m_i = Ar_i$$

$$c_i = Cr_i$$

$$p_i = softmax(u^T m_i)$$

$$o = \sum_{i=1}^{k} p_i c_i$$

$$\hat{u} = o + u$$

Multi-Task Learning

We train our system using multi-task learning as a way of combining conversational data that is naturally associated with external data and other businesses. We use multi-task learning with these tasks:

- NOFACTS task: We expose the model without fact encoder with (S, R) training examples, where S represents the conversation history and R is the response.
- FACTS task: We exposes the full model with $(\{f_1, .., f_k, S\}, R)$ training examples.
- AUTOENCODER task: It is similar to the FACTS task, except that we replace the response with each of the facts.

The tasks FACTS and NOFACTS are representative of how our model is intended to work, but we found that the AUTOENCODER tasks helps inject more factual content into the response.

Multi-Task Learning

The different variants of our multi-task learned system exploits these tasks as follows:

- SEQ2SEQ: This system is trained on task NOFACTS with the 23M general conversation dataset. Since there is only one task, it is not per se a multi-task setting.
- MTASK: This system is trained on two instances of the NOFACTS task, respectively with the 23M general dataset and 1M grounded dataset (but without the facts).
- MTASK-R: This system is trained on the NOFACTS task with the 23M dataset, and the FACTS task with the 1M grounded dataset.

Multi-Task Learning

- MTASK-F: This system is trained on the NOFACTS task with the 23M dataset, and the AUTOENCODER task with the 1M dataset.
- MTASK-RF: This system blends MTASK-F and MTASK-R, as it incorporates 3 tasks: NOFACTS with the 23M general dataset, FACTS with the 1M grounded dataset, and AUTOENCODER again with the 1M dataset.

Multi-Task Learning

We use the same learning technique as (Luong et al., 2015) for multi-task learning.In each batch, all training data is sampled from one task only. For task i we define its mixing ratio value of α_i , and for each batch we select randomly a new task i with probability of $\alpha_i / \sum_j \alpha_j$ and train the system by its training data.

Results

Model	Perplexity General Data	Perplexity Grounded Data
SEQ2SEQ	**55.0**	214.4
SEQ2SEQ-S	125.7	**82.6**
MTASK	57.2	82.5
MTASK-R	**55.1**	**77.6**
MTASK-F	77.3	448.8
MTASK-RF	67.2	97.7

Table 1: Perplexity of different models. SEQ2SEQ-S is a SEQ2SEQ model that is trained on NOFACTS task with 1M grounded dataset (without the facts).

Model	BLEU	Diversity 1-gram	Diversity 2-gram
SEQ2SEQ	0.55	4.14%	14.4%
MTASK	0.79	2.34%	5.9%
MTASK-F	0.38	8.35%	23.1%
MTASK-R	**1.08**	7.08%	21.9%
MTASK-RF	0.58	**8.71%**	**26.0%**

Table 2: BLEU-4 and lexical diversity.

Comparison	Appropriateness				Informativeness			
SEQ2SEQ vs MTASK	**0.530**	**±0.017**	0.470	±0.017	0.501	±0.016	0.499	±0.015
SEQ2SEQ vs MTASK-F	**0.537**	**±0.016**	0.463	±0.017	0.478	±0.015	**0.522**	**±0.015**
SEQ2SEQ vs MTASK-R	0.495	±0.015	0.505	±0.015	0.479	±0.017	**0.521**	**±0.013**
SEQ2SEQ vs MTASK-RF	0.502	±0.015	0.498	±0.014	0.492	±0.013	0.508	±0.013

Table 3: Mean differences in judgments in crowdsourced human evaluation, together with 95% confidence intervals. Differences sum to 1.0. Boldface items are significantly better ($p < 0.05$) than their comparator.

Examples

A: Visiting the celebs at Los Angeles International Airport (LAX) - [...] w/ 70 others B: Nice airport terminal. Have a safe flight.
A: Is that [...] in your photos? It's on my list of places to visit in NYC. B: Don't forget to check out the 5th floor, while you are here, it's a great view.
A: Live right now on [...] Tune in!!!!! B: Listen to Lisa Paige
A: Been craving Chicken Pot Pie-who has the best? Trying [...] at [...] Must be Change of weather! B: Love the pasta trattoria.
A: So [...] is down to one copy of Pound Foolish. I'm curious to see if they are re-ordering it. B: Check out the video feed on 6 and take a picture of the Simpsons on the 3rd floor.
A: I wish [...] would introduce another vegetarian option besides the shroomburger. It's delicious but kind of ridiculous. B: This is the best j.crew in the world. Try the lemonade!
A: Just had an awesome dinner at [...] Great recommendation [...] B: One of my favorite places I've ever been to in NYC. The food is great and the service is lackluster.

Figure 6: Responses produced by our knowledge-grounded neural response generation system (MTASK-RF). Entities marked with [...] have been anonymized to avoid (potentially negative) publicity.

Conclusions

- The model is a largescale, scalable, fully data-driven neural conversation model that effectively exploits external knowledge, and does so without explicit slot filling.
- It generalizes the SEQ2SEQ approach to neural conversation models by naturally combining conversational and non-conversational data through multi-task learning.

Conclusions

- "Neural Generative Question Answering" : The model is built on the encoder-decoder framework for sequence-to-sequence learning, while equipped with the ability to query a knowledge-base.
- "Commonsense Knowledge Aware Conversation": a QA system that has the ability of querying a complex-structured knowledge-base.
- "A Knowledge-Grounded Neural Conversation Model":It generalizes the SEQ2SEQ approach to neural conversation models by naturally combining conversational and non-conversational data through multi-task learning.