1 Introduction

Conversational question answering (CoQA) (Reddy et al., 2018) task combines the properties of machine comprehension and chatbots. In order to predict the answer, we need to understand the question based on dialogue history, and find the answer conditioned on the question and the context paragraph. In this project, we hope to discover the usefulness of structure understanding in CoQA. Specifically, we want to learn the structure from dialogue history to understand the question better.

In the domain of this midterm report, we discuss the setup and implementation details for baseline approaches on CoQA task. Specifically, we would describe in details about each of our baseline models, and evaluate their performances on the CoQA dataset, with further analysis.

2 Baseline Methods

In this section, we introduce a few baseline models we developed for the task of CoQA. Generally, we followed the settings described in (Reddy et al., 2018) and implemented methods of three major categories — conversational models (Sutskever et al., 2014), reading comprehension models (Chen et al., 2017) and their combinations as our baseline approaches. We describe the conversational models in Sec. 2.1 and the reading comprehension models in Sec. 2.2. Finally, we introduce the combined approaches in Sec. 2.3.

2.1 Conversational Model

Seq2seq models (Sutskever et al., 2014) has proved its usefulness in language generation (Zemlyanskiy and Sha, 2018; Serban et al., 2016). In our project, we take a seq2seq model as the baseline of our conversational model. However, the sparsity of language causes many rare words in ground truth answers. Hence we employed the copy mechanism (Vinyals et al., 2015; See et al., 2017) and noted the new model as pointer-generator networks (PG-Net). We will explain the two models in the following.

2.1.1 Sequence to Sequence Model

We use seq2seq model to generate free-form answers conditioned on the context and the question. Specifically, each conversation has a context \( c \), a set of questions \( q = \{q_1, \cdots, q_L\} \) and a set of answers \( a = \{a_1, \cdots, a_L\} \), where \( L \) is the number of turns in conversation. In training, the model learns from predicting \( a_l \) given the question \( q_l \), the conversation history and the context. Assume the history size to be \( m \), we predict the answer \( a_l \) as:

\[
\hat{a}_l = \text{SEQ2SEQ}(x),
\]

where

\[
x = c < q > q_{l-m} < a > a_{l-m} \cdots < q > q_l
\]

For our seq2seq model, we use a bidirectional-LSTM as the encoder, and decode the latent representation using an LSTM with attention mechanism from (Bahdanau et al., 2014).

2.1.2 Pointer Generator Networks

The pointer generator network (PG-Net) (See et al., 2017) is a hybrid between the seq2seq baseline and pointer networks (Vinyals et al., 2015). Besides selecting words from vocabulary, the model has a chance to copy the word from input text at each decoder time step.

Let us define a generation probability \( p_g \), which is the probability for decoder to generate a word from vocabulary. The new word is generated by sampling from \( p_v \), where \( p_v \) is a categorical distribution over all vocabularies. Or, instead of sampling from \( p_v \), the model can copy a word from
the input text. The probability of copying the word follows the attention weight distribution $\alpha$. Note that $\alpha$ is also a categorical distribution over words in input sequence, where $\alpha_{ij}$ represents how well the decoder state at time $i$ matches the encoder representation at time $j$. And $\sum_j \alpha_{ij} = 1$. (The detail of attention weights computation can be found in (Bahdanau et al., 2014)). Now the output word probability distribution becomes:

$$p(w) = p_g p_c(w) + (1 - p_g) \sum_{j=1}^T \mathbb{I}(x_j == w) \alpha_{ij}$$

### 2.2 Reading Comprehension Model

State-of-the-art reading comprehension models focus on finding a text span that matches the question best as the answer to the question given the context. In this case, such system might not be able to handle the questions whose answers do not overlap with the context story. While having those limitation, such model enables the efficient learning as they constraint the answer space to be a finite set, instead of predicting over the infinite space of possible answers.

To this end, we follow (Chen et al., 2017) to implement the state-of-the-art Document Reader (DrQA) model as one of the baselines. To be concise, DrQA first encodes a paragraph as a sequence of feature vectors, which consists of word embeddings, binary exact matches of question words, token feature and aligned question embeddings (see (Chen et al., 2017) for details). Such sequence of feature vectors are then used as input to a bidirectional LSTM (Hochreiter and Schmidhuber, 1997) for contextual feature extraction. We note a sequence of such story features as $\{p_1,...,p_m\}$.

Next, question features are extracted in a similar fashion and aggregated into one vector $q$ via weighted combination of each word. Based on the bilinearly combined question embedding and story embeddings at different position, a multilayer perception (MLP) model is used to predict the probability of start each position as start span $(s)$ and end $(e)$ span:

$$P_s(i) \propto \exp (p_i W_s q)$$  

$$P_e(i) \propto \exp (p_i W_e q)$$

To accomodate the training of DrQA, we select the text span which has the highest lexical overlap (Macro F1 score) of the original answers in the context story as the gold answer. If any answer word does not appear in the context story, an unknown token is used as gold answer. Each question is prepend to its past questions and answers as conversational history.

### 2.3 Combined Models

As discussed above, we have noted the fact that reading comprehension based model can not really accommodate the scenario where the original answer words are not overlapping with the context story. To this end, a straight-forward extension would be combining them with conversational models to generate free-from answers. We would expect a model to first predict a text span and then refine the predicted answer.

### 3 Experiments

In this section, we will describe the experiments setup and compare the performance of our baseline models seq2seq (Sec. 2.1.1), PG-Net (Sec. 2.1.2), DrQA (Sec. 2.2) and the combined model (Sec. 2.3).

#### 3.1 Experiment Setup

We implement all baseline approaches and evaluation metrics based on the original implemented from (Reddy et al., 2018). It contains Seq2Seq and PG-Net models based on OpenNMT toolkit. We also adopted the implementation of DrQA model from (Chen et al., 2017).

Following SQuAD (Rajpurkar et al., 2016), we use macro-average F1 score of word overlap, as well as the exact match (EM) score as our main evaluation metrics.

We employed pretrained 300 dimension GloVe (Pennington et al., 2014) embedding and fine-tuned the embedding. And we use the Adam optimizer (Kingma and Ba, 2014) for training.

#### 3.2 Results and Analysis

The performance on development set are shown in Table 1.

Based on the performance on development set, seq2seq model performs the worst. By looking into the prediction results, seq2seq model is likely to predict common words appeared in training set such as “yes”. PG-Net alleviates this problem by assigning the probability of copying words from the input sequence. Although PG-Net achieves a huge performance gain comparing to seq2seq
Table 1: F1 score on development set. (Due to the limited time we have, the combined model haven’t finished training and therefore result is not available yet.)

<table>
<thead>
<tr>
<th>Models</th>
<th>F1 score</th>
<th>Exact Match</th>
</tr>
</thead>
<tbody>
<tr>
<td>seq2seq</td>
<td>21.5</td>
<td>17.7</td>
</tr>
<tr>
<td>PG-Net</td>
<td>40.2</td>
<td>33.4</td>
</tr>
<tr>
<td>DrQA</td>
<td>53.6</td>
<td>44.3</td>
</tr>
<tr>
<td>combined model</td>
<td>–</td>
<td>–</td>
</tr>
</tbody>
</table>

model, DrQA model performs better than PG-Net. One reason could be that DrQA predicts the start and the end of the answer span instead of generating free-form answers, which makes it easier for the model to achieve higher score. Also, PG-Net memorizes the context and question jointly to generate the answer, while DrQA encodes the context and question separately, so as it can find answers from context given the question embedding. The performance of combined model is not finished yet, but we expect that it will achieve the best performance among all our baseline models.

4 Future Plan

From this point on, we would follow our initial plan to investigate whether learning structured entity graph over the conversational history would help a learning system to better answer the question. In a very recent related work, (Huang et al., 2018) show that it is crucial to have a long range reasoning over both the conversational histories and the context story, which improved previous state-of-the-art systems by a large marigin (~ 10%). This seems supporting our research hypothesis about the importance of understanding the conversation. We plan to experiment more along this direction and we expect to improve our current system with this.

References


