GRAPHFLOW: Exploiting Conversation Flow with Graph Neural Networks for Conversational Machine Comprehension

Yu Chen [1] **Lingfei Wu** [2] **Mohammed J. Zaki** [1]

Abstract

Conversational machine reading comprehension (MRC) has proven significantly more challenging compared to traditional MRC since it requires better utilization of conversation history. However, most existing approaches do not effectively capture conversation history and thus have trouble handling questions involving coreference or ellipsis. We propose a novel graph neural network (GNN) based model, namely GRAPHFLOW, which captures conversational flow in the dialog. Specifically, we first propose a new approach to dynamically construct a question-aware context graph from passage text at each turn. We then present a novel flow mechanism to model the temporal dependencies in the sequence of context graphs. The proposed GRAPHFLOW model shows superior performance compared to existing state-of-the-art methods. For instance, GRAPHFLOW outperforms two recently proposed models on the CoQA benchmark dataset: FLOWQA by 2.3% and SDNet by 0.7% on F1 score, respectively.

1. Introduction

Recent years have observed a surge of interest in conversational machine reading comprehension (MRC). Unlike the traditional setting of MRC that requires answering a single question given a passage (aka context), the conversational MRC task is to answer the current question in a conversation given a passage and the previous questions and answers. The goal of this task is to mimic real-world situations where humans seek information in a conversational manner.

Despite the success existing works have achieved on traditional MRC (e.g., SQuAD (Rajpurkar et al., 2016)), conversational MRC has proven significantly more challenging when the conversations are incorporated into the MRC task. During a conversation (Reddy et al., 2018; Choi et al., 2018), it has been observed that shifts of focus happen frequently and many questions refer back to the conversation history via either coreference or ellipsis. We model *conversation flow* as a sequence of latent states in the dialog and learn important latent states associated with these shifts of focus.

To cope with the above challenges, we propose GRAPHFLOW, a Graph Neural Network (GNN) based model for conversational MRC. As shown in Fig. 1, GRAPHFLOW consists of three components, Encoding layer, Reasoning layer, and Prediction layer. The Encoding layer encodes conversation history and the context text that aligns question embeddings. The Reasoning layer dynamically constructs a question-aware context graph at each turn, and then applies GNNs to process the sequence of context graphs. In particular, the graph node embedding outputs of the reasoning process at the previous turn are used as a starting state when reasoning at the current turn, which is closer to how humans perform reasoning in a conversational setting, compared to existing approaches. The prediction layer predicts the answers based on the matching scores of the question embedding and the context graph node embeddings per turn.

2. Graph-Flow Approach

2.1. Encoding Layer

We denote the context as C which is a sequence of words $\{c_1, c_2, ..., c_m\}$ and the question at the i-th turn as Q_i which is a sequence of words $\{q_1^{(i)}, q_2^{(i)}, ..., q_n^{(i)}\}$. The details of encoding the question and context are given next.

Pretrained word embeddings We use 300-dim GloVe (Pennington et al., 2014) embeddings as well as 1024-dim BERT (Devlin et al., 2018) embeddings to embed each word in the context and the question. Following (Zhu et al., 2018), we pre-compute BERT embeddings for each word using a weighted sum of BERT layer outputs.

Aligned question embeddings Following (Lee et al., 2016) and recent work, for each context word c_j at the i-th turn, we incorporate an aligned question embedding $f_{align}(c_j^{(i)}) = \sum_k a_{j,k}^{(i)} \mathbf{g}_k^{Q_i}$ where $\mathbf{g}_k^{Q_i}$ is the GloVe embedding of question

[1]Computer Science Department, Rensselaer Polytechnic Institute, Troy, NY, USA [2]IBM Research, Yorktown Heights, NY, USA. Correspondence to: Lingfei Wu <lwu@email.wm.edu>.

Presented at the ICML 2019 Workshop on Learning and Reasoning with Graph-Structured Data

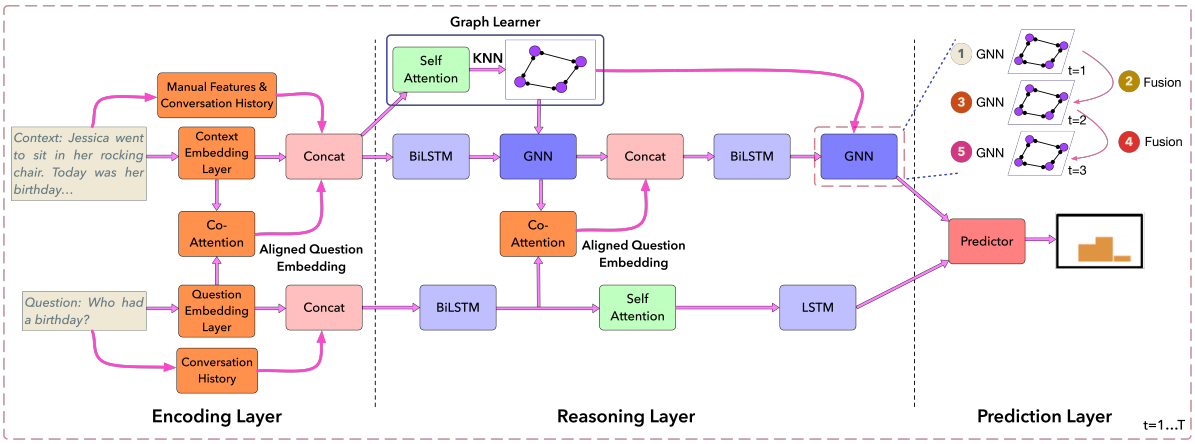

Figure 1. Overall architecture of the proposed model. Best viewed in color.

word $q_k^{(i)}$ and $a_{j,k}^{(i)}$ is an attention score between context word c_j and question word $q_k^{(i)}$. Here we define the attention score $a_{j,k}^{(i)}$ as,

$$a_{j,k}^{(i)} \propto \exp(\text{ReLU}(\mathbf{W}\mathbf{g}_j^C)^T \text{ReLU}(\mathbf{W}\mathbf{g}_k^{Q_i})) \quad (1)$$

where $\mathbf{W} \in \mathbb{R}^{d \times d}$ is a trainable model parameter, d is the hidden state size, and $\mathbf{g}_j^C$ is the GloVe embedding of context word c_j. To simplify notation, we denote the above attention mechanism as $\text{Align}(\mathbf{A}, \mathbf{B}, \mathbf{C})$, meaning that an attention matrix is computed between two sets of vectors $\mathbf{A}$ and $\mathbf{B}$, which is later used to get a linear combination of vector set $\mathbf{C}$. Hence we can reformulate the above alignment as $f_{\text{align}}(C^{(i)}) = \text{Align}(\mathbf{g}^C, \mathbf{g}^{Q_i}, \mathbf{g}^{Q_i})$.

Linguistic features Following previous works (Chen et al., 2017; Huang et al., 2018; Zhu et al., 2018), for each context word, we also encode linguistic features to a vector $f_{\text{ling}}(c_j^{(i)})$ concatenating 12-dim POS (part-of-speech) embedding, 8-dim NER (named entity recognition) embedding and a 3-dim exact matching vector indicating whether the context word appears in Q_i.

Conversation history Following (Choi et al., 2018), we utilize conversation history by concatenating a feature vector $f_{\text{ans}}(c_j^{(i)})$ encoding previous N answer locations to the context word embeddings. In addition, we prepend previous N question-answer pairs to the current question and concatenate a 3-dim turn marker embedding $f_{\text{turn}}(q_k^{(i)})$ to each word vector in the augmented question to indicate which turn it belongs to (e.g., i indicates the previous i-th turn).

In summary, at the i-th turn in a conversation, each context word c_j is encoded by a vector $\mathbf{w}_{c_j}^{(i)}$ which is a concatenation of $\mathbf{g}_j^C$, BERT_j^C, $f_{\text{align}}(c_j^{(i)})$, $f_{\text{ling}}(c_j^{(i)})$ and $f_{\text{ans}}(c_j^{(i)})$. And each question word $q_k^{(i)}$ is encoded by a vector $\mathbf{w}_k^{Q_i}$ which is a concatenation of $\mathbf{g}_k^{Q_i}$, $\text{BERT}_k^{Q_i}$ and $f_{\text{turn}}(q_k^{(i)})$. We denote $\mathbf{W}_C^{(i)}$ and $\mathbf{W}^{Q_i}$ as a sequence of context word vectors $\mathbf{w}_{c_j}^{(i)}$ and question word vectors $\mathbf{w}_k^{Q_i}$, respectively at the i-th turn.

2.2. Reasoning Layer

2.2.1. Question Understanding

For each question Q_i, we apply a BiLSTM (Hochreiter & Schmidhuber, 1997) to the raw question embeddings $\mathbf{W}^{Q_i}$ to obtain contextualized embeddings $\mathbf{Q}_i \in \mathbb{R}^{d \times n}$.

$$\mathbf{Q}_i = \mathbf{q}_1^{(i)}, ..., \mathbf{q}_n^{(i)} = \text{BiLSTM}(\mathbf{W}^{Q_i}) \quad (2)$$

Each question is then represented as a weighted sum of word vectors in the question via a self-attention mechanism.

$$\tilde{\mathbf{q}}^{(i)} = \sum_k a_k^{(i)} \mathbf{q}_k^{(i)}, \quad \text{where } a_k^{(i)} \propto \exp(\mathbf{w}^T \mathbf{q}_k^{(i)}) \quad (3)$$

where $\mathbf{w}$ is a d-dim trainable weight.
Finally, we encode question history sequentially in turns with a LSTM to generate history-aware question vectors.

$$\mathbf{p}_1, ..., \mathbf{p}_T = \text{LSTM}(\tilde{\mathbf{q}}^{(1)}, ..., \tilde{\mathbf{q}}^{(T)}) \quad (4)$$

The output hidden states of the LSTM network $\mathbf{p}_1, ..., \mathbf{p}_T$ will be used for predicting answers.

2.2.2. Graph Learner

We now introduce how to dynamically build a weighted graph to model semantic relationships among context words at each turn in a conversation. To this end, we first apply an attention mechanism to the context representations $\mathbf{W}_C^{(i)}$ (which additionally incorporate both question information and conversation history as described in Section 2.1) at the i-th turn to compute an attention matrix $\mathbf{A}_C^{(i)}$, serving as a weighted adjacency matrix for the context graph, defined as,

$$\mathbf{A}_C^{(i)} = \text{ReLU}(\mathbf{U}\mathbf{W}_C^{(i)})^T \text{ReLU}(\mathbf{U}\mathbf{W}_C^{(i)}) \quad (5)$$

where $\mathbf{U}$ is a $d \times d_c$ trainable weight and d_c is the embedding size of $\mathbf{w}_{c_j}^{(i)}$.

Considering that a fully connected context graph is not only computationally expensive but also makes little sense for reasoning, we proceed to extract a sparse graph from $\mathbf{A}_C^{(i)}$ via a KNN-style strategy where we only keep the K nearest neighbors (including itself) for each context node and apply a softmax function to these selected adjacency matrix elements to get a sparse and normalized adjacency matrix $\widetilde{\mathbf{A}}_C^{(i)}$.

$$\widetilde{\mathbf{A}}_C^{(i)} = \text{softmax}(\text{topk}(\mathbf{A}_C^{(i)})) \tag{6}$$

2.2.3. GRAPH-FLOW

Given the context graphs constructed by the Graph Learner, we propose a novel *Graph-Flow* (GF) mechanism to sequentially process a sequence of context graphs. Readers can think that it is analogous to an RNN-style structure where the main difference is that each element in a sequence is not a data point, but instead a graph. As we advance in a sequence of graphs, we process each graph using a GNN and the output will be used when processing the next graph.

The details of the GF mechanism are as follows. At the i-th turn, before we apply a GNN to the context graph $\mathcal{G}_i$, we initialize context node embeddings by fusing both the original context information $\mathbf{C}_i^{l-1}$ and the updated context information at the previous turn $\mathbf{C}_{i-1}^{l}$ via a fusion function.

$$\begin{aligned} \mathbf{C}_i^l &= \text{GNN}(\bar{\mathbf{C}}_i^{l-1}, \widetilde{\mathbf{A}}_C^{(i)}) \\ \bar{\mathbf{C}}_{i,j}^{l-1} &= \text{Fuse}(\mathbf{C}_{i,j}^{l-1}, \mathbf{C}_{i-1,j}^{l}) \end{aligned} \tag{7}$$

where l is the GF layer index. Note that we can stack multiple GF layers to enhance the performance if necessary.

As a result, the graph node embedding outputs of the reasoning process at the previous turn are used as a starting state when reasoning at the current turn. Note that we set $\bar{\mathbf{C}}_0^{l-1} = \mathbf{C}_0^{l-1}$ as we will not incorporate any historical information at the first turn.

We use Gated Graph Neural Networks (GGNN) (Li et al., 2015) as our GNN module. When running GGNN, the aggregated neighborhood information for each node is computed as a weighted sum of its neighboring node embeddings where the weights come from the normalized adjacency matrix $\widetilde{\mathbf{A}}_C^{(i)}$. The fusion function is designed as a gated sum of two information sources,

$$\begin{aligned} \text{Fuse}(\mathbf{a}, \mathbf{b}) &= \mathbf{z} * \mathbf{a} + (1 - \mathbf{z}) * \mathbf{b} \\ \mathbf{z} &= \sigma(\mathbf{W}_z[\mathbf{a}; \mathbf{b}; \mathbf{a} * \mathbf{b}; \mathbf{a} - \mathbf{b}] + \mathbf{b}_z) \end{aligned} \tag{8}$$

where σ is a sigmoid function and $\mathbf{z}$ is a gating vector.

To simplify notation, we denote the GF mechanism as $\mathbf{C}^l = \text{Graph-Flow}(\mathbf{C}^{l-1}, \widetilde{\mathbf{A}}_C)$ which takes as input the old graph node embeddings $\mathbf{C}^{l-1}$ and the normalized adjacency matrix $\widetilde{\mathbf{A}}_C$, and updates the graph node embeddings.

2.2.4. MULTI-LEVEL GRAPH REASONING

While a GNN is responsible for modeling the global interactions among context words, modeling local interactions among consecutive context words is also important for the task. Therefore, before feeding the context word representations to a GNN, we first apply a BiLSTM to the context words, that is, $\mathbf{C}_i^0 = \text{BiLSTM}(\mathbf{W}_C^{(i)})$, and we then use the output $\mathbf{C}_i^0$ as the initial context node embedding. Inspired by recent work (Wang et al., 2018) on modeling the context with different levels of granularity, we choose to apply one GF layer on low level representations of the context and another GF layer on high level representations of the context, as formulated in the following.

$$\begin{aligned} \mathbf{C}^1 &= \text{Graph-Flow}(\mathbf{C}^0, \widetilde{\mathbf{A}}_C) \\ \mathbf{H}_i^Q &= [\mathbf{Q}_i; \mathbf{g}^{Q_i}; \text{BERT}^{Q_i}] \\ \mathbf{H}_i^C &= [\mathbf{C}_i^1; \mathbf{g}^C; \text{BERT}^C] \\ f_{\text{align}}^2(C^{(i)}) &= \text{Align}(\mathbf{H}_i^C, \mathbf{H}_i^Q, \mathbf{Q}_i)] \\ \widetilde{\mathbf{C}}_i^1 &= \text{BiLSTM}([\mathbf{C}_i^1; f_{\text{align}}^2(C^{(i)})]) \\ \mathbf{C}^2 &= \text{Graph-Flow}(\widetilde{\mathbf{C}}^1, \widetilde{\mathbf{A}}_C) \end{aligned} \tag{9}$$

2.3. Prediction Layer

Following (Huang et al., 2018; Zhu et al., 2018), we use the same answer span selection method to predict the start and end probabilities $P_{i,j}^S$ and $P_{i,j}^E$ of the j-th context word for the i-th question. We additionally train a classifier to handle unanswerable questions or questions whose answers are not text spans in the context. A detail restatement of the answer span selection method can be found in Appendix A.

3. Experiments

In this section, we conduct an extensive evaluation of our proposed model against state-of-the-art conversational MRC models. We use two popular benchmarks, described below.

3.1. Data and Metrics

The CoQA data contains 127k questions with answers, obtained from 8k conversations. In CoQA, answers are in free-form and hence are not necessarily text spans from the context. The QuAC data contains 98k questions with answers, obtained from 13k conversations. All the answers in QuAC are text spans from the context.

The main evaluation metric is F1 score which is the harmonic mean of precision and recall at word level between the predicted answer and ground truth. In addition, for QuAC the Human Equivalence Score (i.e., HEQ-Q and

Table 1 Model and human performance (% in F1 score) on the CoQA test set.

	Child.	Liter.	Mid-High.	News	Wiki	Reddit	Science	Overall
PGNet (See et al., 2017)	49.0	43.3	47.5	47.5	45.1	38.6	38.1	44.1
DrQA (Chen et al., 2017)	46.7	53.9	54.1	57.8	59.4	45.0	51.0	52.6
DrQA+PGNet (Reddy et al., 2018)	64.2	63.7	67.1	68.3	71.4	57.8	63.1	65.1
BiDAF++ (Yatskar, 2018)	66.5	65.7	70.2	71.6	72.6	60.8	67.1	67.8
FLOWQA (Huang et al., 2018)	73.7	71.6	76.8	79.0	80.2	67.8	76.1	75.0
SDNet (Zhu et al., 2018)	75.4	73.9	77.1	**80.3**	**83.1**	69.8	76.8	76.6
GRAPHFLOW	**77.1**	**75.6**	**77.5**	79.1	82.5	**70.8**	**78.4**	**77.3**
Human (Reddy et al., 2018)	90.2	88.4	89.8	88.6	89.9	86.7	88.1	88.8

Table 2 Model and human performance (in %) on the QuAC test set.

	F1	HEQ-Q	HEQ-D
BiDAF++ (Yatskar, 2018)	60.1	54.8	4.0
FLOWQA (Huang et al., 2018)	64.1	59.6	**5.8**
GRAPHFLOW	**64.9**	**60.3**	5.1
Human (Choi et al., 2018)	80.8	100	100

HEQ-D) is used to judge whether a system performs as well as an average human. Please refer to (Reddy et al., 2018; Choi et al., 2018) for details of these metrics.

3.2. Model Comparison

As shown in Table 1 and Table 2, our model consistently outperforms these state-of-the-art baselines in terms of F1 score. In particular, GRAPHFLOW yields improvement over all existing models on both datasets by at least +0.7% F1 on CoQA and +0.8% F1 on QuAC, respectively. Compared with FLOWQA which is also based on the flow idea, our model improves F1 by 2.3% on CoQA and 0.8% on QuAC, which demonstrates the superiority of our GF mechanism over the *Integration-Flow* mechanism. Compared with SDNet which relies on sophisticated inter-attention and self-attention mechanisms, our model improves F1 by 0.7% on CoQA (They did not report the results on QuAC.).

3.3. Ablation Study

Table 3 Ablation study: model performance (in %) on the CoQA dev. set.

	F1
GRAPHFLOW (2-His)	**78.3**
– PreQues	78.2
– PreAns	77.7
– PreAnsLoc	76.6
– BERT	70.2
– GF	68.8
– TempConn	69.9
GRAPHFLOW (1-His)	78.2
GRAPHFLOW (0-His)	76.7

We conduct an extensive ablation study to further investigate the performance impact of different components in our model as shown in Table 3. We find that the pretrained BERT embedding (i.e., – BERT) has the most impact on the performance, which again demonstrates the power of large-scale pretrained language models. Our proposed GF mechanism (i.e., – GF) also contributes significantly to the model performance (i.e., improves F1 score by 1.4%). In addition, within the GF layer, both the GNN part (i.e., 1.1% F1) and the temporal connection part (i.e., 0.3% F1) contribute to the results. We also notice that explicitly adding conversation history to the current turn helps the model performance by comparing GRAPHFLOW (2-His), GRAPHFLOW (1-His) and GRAPHFLOW (0-His). We can see that the previous answer information (i.e., – PreAns) is more crucial than the previous question information (i.e., – PreQues). And among many ways to use the previous answer information, directly marking previous answer locations (i.e., – PreAnsLoc) seems to be the most effective. We conjecture this is partially because the turn transitions in a conversation are usually smooth and marking the previous answer locations helps the model better identify relevant context chunks for the current question.

4. Conclusion

We proposed a novel GNNs-based model, namely GRAPHFLOW, for conversational MRC which carries over the reasoning output throughout a conversation. On two recently released conversational MRC benchmarks, our proposed model achieves superior results over previous approaches.

In the future, we would like to investigate more effective ways of automatically learning graph structures from free text and modeling temporal connections between sequential graphs.

Acknowledgements

This work is supported by IBM Research AI through the IBM AI Horizons Network. We thank the anonymous reviewers for their constructive suggestions.

References

Chen, D., Fisch, A., Weston, J., and Bordes, A. Reading wikipedia to answer open-domain questions. *arXiv preprint arXiv:1704.00051*, 2017.

Cho, K., van Merrienboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *EMNLP*, pp. 1724–1734, 2014.

Choi, E., He, H., Iyyer, M., Yatskar, M., Yih, W.-t., Choi, Y., Liang, P., and Zettlemoyer, L. Quac: Question answering in context. *arXiv preprint arXiv:1808.07036*, 2018.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Hochreiter, S. and Schmidhuber, J. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

Huang, H.-Y., Choi, E., and Yih, W.-t. Flowqa: Grasping flow in history for conversational machine comprehension. *arXiv preprint arXiv:1810.06683*, 2018.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Lee, K., Salant, S., Kwiatkowski, T., Parikh, A., Das, D., and Berant, J. Learning recurrent span representations for extractive question answering. *arXiv preprint arXiv:1611.01436*, 2016.

Li, Y., Tarlow, D., Brockschmidt, M., and Zemel, R. Gated graph sequence neural networks. *arXiv preprint arXiv:1511.05493*, 2015.

Pennington, J., Socher, R., and Manning, C. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pp. 1532–1543, 2014.

Rajpurkar, P., Zhang, J., Lopyrev, K., and Liang, P. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*, 2016.

Reddy, S., Chen, D., and Manning, C. D. Coqa: A conversational question answering challenge. *arXiv preprint arXiv:1808.07042*, 2018.

See, A., Liu, P. J., and Manning, C. D. Get to the point: Summarization with pointer-generator networks. *arXiv preprint arXiv:1704.04368*, 2017.

Wang, W., Yan, M., and Wu, C. Multi-granularity hierarchical attention fusion networks for reading comprehension and question answering. *arXiv preprint arXiv:1811.11934*, 2018.

Yatskar, M. A qualitative comparison of coqa, squad 2.0 and quac. *arXiv preprint arXiv:1809.10735*, 2018.

Zhu, C., Zeng, M., and Huang, X. Sdnet: Contextualized attention-based deep network for conversational question answering. *arXiv preprint arXiv:1812.03593*, 2018.

A. Prediction Layer

Following (Huang et al., 2018; Zhu et al., 2018), we predict answer spans by computing the start and end probabilities $P^S_{i,j}$ and $P^E_{i,j}$ of the j-th context word for the i-th question, formulated as,

$$\begin{aligned} P^S_{i,j} &\propto \exp({\mathbf{c}^2_{i,j}}^T \mathbf{W}_S \mathbf{p}_i) \\ \widetilde{\mathbf{p}}_i &= \text{GRU}(\mathbf{p}_i, \sum_j P^S_{i,j} \mathbf{c}^2_{i,j}) \\ P^E_{i,j} &\propto \exp({\mathbf{c}^2_{i,j}}^T \mathbf{W}_E \widetilde{\mathbf{p}}_i) \end{aligned} \tag{10}$$

where $\mathbf{W}_S$ and $\mathbf{W}_E$ are $d \times d$ trainable weights and GRU is a Gated Recurrent Unit (Cho et al., 2014).

We additionally train a classifier to handle unanswerable questions or questions whose answers are not text spans in the context. We design different classifiers for the two benchmarks CoQA and QuAC as CoQA contains questions with abstractive answers but QuAC does not. For the CoQA benchmark, we train a multi-class classifier which classifies a question into one of the four categories including "unknown", "yes", "no" and "other". We do text span prediction only if the question type is "other". For the QuAC benchmark, we train three separate classifiers to handle three question classification tasks including a binary classification task (i.e., "unknown") and two multi-class classification tasks (i.e., "yes/no" and "followup"). The classifier is defined as,

$$\begin{aligned} \widetilde{\mathbf{C}}^2_i &= [f_{\text{mean}}(\mathbf{C}^2_i); f_{\max}(\mathbf{C}^2_i)] \\ P^C_i &= \sigma(f_c(\mathbf{p}_i) \widetilde{\mathbf{C}}^{2T}_i) \end{aligned} \tag{11}$$

where f_c is a linear layer for binary classification and a dense layer for multi-class classification, which maps a d-dim vector to a $(\text{num\_class} \times 2d)$-dim vector. Further, σ is a sigmoid function for binary classification and a softmax function for multi-class classification. We use $\widetilde{\mathbf{C}}^2_i$ to represent the whole context at the i-th turn which is a concatenation of average pooling and max pooling outputs of $\mathbf{C}^2_i$.

A.0.1. TRAINING

For training, the goal is to minimize the cross entropy loss of both text span prediction (if the question requires it) and question type prediction. The cross entropy of text span prediction is defined as,

$$\mathcal{L}_S = -\sum_i I_i^S (log(P_{i,s_i}^S) + log(P_{i,e_i}^E)) \tag{12}$$

where I_i^S indicates whether this question requires text span prediction, and s_i and e_i are the ground-truth start and end positions of the answer span for the i-th question.

As aforementioned, we train a single classifier for question type prediction on CoQA and three separate classifiers for question type prediction on QuAC. Therefore, the loss of question type prediction is defined differently for the two datasets as the following,

$$\mathcal{L}_C^{\text{CoQA}} = -\sum_i \log P_{i,t_i}^C \tag{13}$$

where t_i indicates the question type for the i-th question.

$$\mathcal{L}_C^{\text{QuAC}} = -\sum_i \Big\{ t_i^U \log P_i^U + (1 - t_i^U) \log(1 - P_i^U) + \log P_{i,t_i^Y}^Y + \log P_{i,t_i^F}^F \Big\} \tag{14}$$

where t_i^U, t_i^Y and t_i^F indicate the ground-truth labels of the "unknown", "yes/no" and "followup" prediction tasks for i-th question, and P_i^U, P_i^Y and P_i^F are the corresponding probability predictions.

Thus, the training losses for CoQA and QuAC are $\mathcal{L}_S + \mathcal{L}_C^{CoQA}$ and $\mathcal{L}_S + \mathcal{L}_C^{QuAC}$, respectively.

A.0.2. PREDICTION

During inference, for CoQA, we do text span prediction only if span probability is the largest; otherwise, the answer is "unknown", "yes" or "no" depending on which one has the largest probability. For QuAC, we do text span prediction only if P_i^U is no larger than a certain threshold[1]; otherwise, the question is unanswerable.

B. Model Settings

We keep and fix the GloVe vectors for those words that appear more than 5 times in the training set. The size of all hidden layers is set to 300. When constructing context graphs, the neighborhood size is set to 10. The number of GNN hops is set to 5 for CoQA and 3 for QuAC. During training, we apply dropout after the embedding layers (0.3 for GloVe and 0.4 for BERT). A dropout rate of 0.3 is also applied after the output of all RNN layers. We use Adamax (Kingma & Ba, 2014) as the optimizer and the learning rate is set to 0.001. We reduce the learning rate by a factor of 0.5 if the validation F1 score has stopped improving every one epoch. We stop the training when no improvement is seen for 10 consecutive epochs. We clip the gradient at length 10. We batch over dialogs and the batch size is set to 1. When augmenting the current turn with conversation history, we only consider the previous two turns. When doing text span prediction, the span is constrained to have a maximum length of 12 for CoQA and 35 for QuAC. All these hyper-parameters are tuned on the development set.

[1]We use 0.3 in our experiments as this maximizes the F1 score on the development set.

C. Effects of Parameter Tuning

We study the effects of various hyperparameter choices in GRAPHFLOW regarding the GF component, such as the number of GNN hops and the KNN neighborhood size. The number of GNN hops controls how far node information can be propagated in a graph. The KNN neighborhood size controls the sparsity of the constructed graph. Fig. 2 and Fig. 3 show the F1 score on the QuAC dev. set with various values for the number of GNN hops and the KNN neighborhood size, respectively. The default values for the number of GNN hops and the KNN neighborhood size are 3 and 10, respectively. As we can see, both of the two hyperparameters have significant impacts on the model performance. When reaching an optimal value, further increasing the number of GNN hops or KNN neighborhood size does not help the model performance.

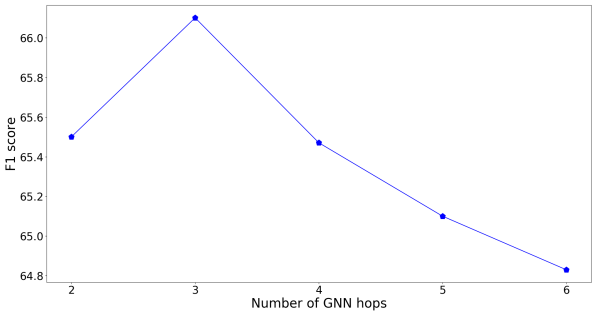

Figure 2. Effect of number of GNN hops on the QuAC dev. set.

D. Interpretability Analysis

Here we visualize the memory bank (i.e., an m by d matrix) which stores the hidden representations (and thus reasoning output) of the context throughout a conversation. While directly visualizing the hidden representations is difficult, thanks to the flow-based mechanism introduced into our model, we instead visualize the changes of hidden representations of context words between consecutive turns. We

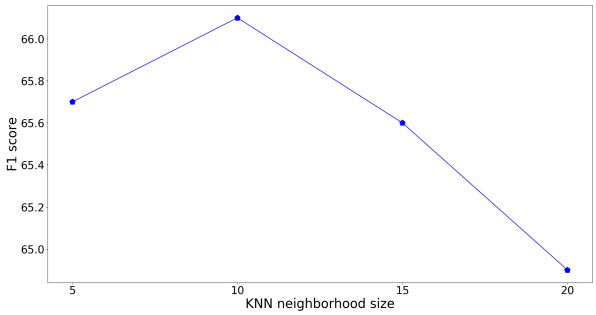

Figure 3. Effect of KNN neighborhood size on the QuAC dev. set.

expect that the most changing parts of the context should be those which are relevant to the questions being asked and therefore should probably be able to indicate shifts of the focus in a conversation.

Following (Huang et al., 2018), we visualize this by computing the cosine similarity of the hidden representations of the same context words at consecutive turns, and then highlight the words that have small cosine similarity scores (i.e., change more significantly). Note that for better visualization, we apply an attention threshold of 0.3 to highlight only the dramatically changing context words. Fig. 4 highlights the most changing context words between consecutive turns in a conversation from the CoQA dev. set. As we can see, the hidden representations of context words which are relevant to the consecutive questions are changing most and thus highlighted most. We suspect this is in part because when the focus shifts, the model finds out the context chunks relevant to the previous turn become less important but those relevant to the current turn become more important. Therefore, the memory updates in these regions are the most active. Obviously, this makes the model easier to answer follow-up questions. As we observe in our visualization experiments, in conversations extensively involving coreference or ellipsis, our model can still perform reasonably well.

Q1: Who went to the farm? -> Q2: Why?

Billy went to the farm to buy some beef for his brother 's birthday . When he arrived there he saw that all six of the cows were sad and had brown spots . The cows were all eating their breakfast in a big grassy meadow . He thought that the spots looked very strange so he went closer to the cows to get a better look . When he got closer he also saw that there were five white chickens sitting on the fence . The fence was painted blue and had some dirty black spots on it . Billy wondered where the dirty spots had come . Soon he got close to the chickens and they got scared . All five chickens flew away and went to eat some food . After Billy got a good look at the cows he went to the farmer to buy some beef . The farmer gave him four pounds of beef for ten dollars . Billy thought that it was a good deal so he went home and cooked his brother dinner . His brother was very happy with the dinner . Billy 's mom was also very happy .

Q2: Why? -> Q3: For what?

Billy went to the farm to buy some beef for his brother 's birthday . When he arrived there he saw that all six of the cows were sad and had brown spots . The cows were all eating their breakfast in a big grassy meadow . He thought that the spots looked very strange so he went closer to the cows to get a better look . When he got closer he also saw that there were five white chickens sitting on the fence . The fence was painted blue and had some dirty black spots on it . Billy wondered where the dirty spots had come . Soon he got close to the chickens and they got scared . All five chickens flew away and went to eat some food . After Billy got a good look at the cows he went to the farmer to buy some beef . The farmer gave him four pounds of beef for ten dollars . Billy thought that it was a good deal so he went home and cooked his brother dinner . His brother was very happy with the dinner . Billy 's mom was also very happy .

Q3: For what? -> Q4: How many cows did he see there?

Billy went to the farm to buy some beef for his brother 's birthday . When he arrived there he saw that all six of the cows were sad and had brown spots . The cows were all eating their breakfast in a big grassy meadow . He thought that the spots looked very strange so he went closer to the cows to get a better look . When he got closer he also saw that there were five white chickens sitting on the fence . The fence was painted blue and had some dirty black spots on it . Billy wondered where the dirty spots had come . Soon he got close to the chickens and they got scared . All five chickens flew away and went to eat some food . After Billy got a good look at the cows he went to the farmer to buy some beef . The farmer gave him four pounds of beef for ten dollars . Billy thought that it was a good deal so he went home and cooked his brother dinner . His brother was very happy with the dinner . Billy 's mom was also very happy .

Q4: How many cows did he see there? -> Q5: Did they have spots?

Billy went to the farm to buy some beef for his brother 's birthday . When he arrived there he saw that all six of the cows were sad and had brown spots . The cows were all eating their breakfast in a big grassy meadow . He thought that the spots looked very strange so he went closer to the cows to get a better look . When he got closer he also saw that there were five white chickens sitting on the fence . The fence was painted blue and had some dirty black spots on it . Billy wondered where the dirty spots had come . Soon he got close to the chickens and they got scared . All five chickens flew away and went to eat some food . After Billy got a good look at the cows he went to the farmer to buy some beef . The farmer gave him four pounds of beef for ten dollars . Billy thought that it was a good deal so he went home and cooked his brother dinner . His brother was very happy with the dinner . Billy 's mom was also very happy .

Q5: Did they have spots? -> Q6: What color?

Billy went to the farm to buy some beef for his brother 's birthday . When he arrived there he saw that all six of the cows were sad and had brown spots . The cows were all eating their breakfast in a big grassy meadow . He thought that the spots looked very strange so he went closer to the cows to get a better look . When he got closer he also saw that there were five white chickens sitting on the fence . The fence was painted blue and had some dirty black spots on it . Billy wondered where the dirty spots had come . Soon he got close to the chickens and they got scared . All five chickens flew away and went to eat some food . After Billy got a good look at the cows he went to the farmer to buy some beef . The farmer gave him four pounds of beef for ten dollars . Billy thought that it was a good deal so he went home and cooked his brother dinner . His brother was very happy with the dinner . Billy 's mom was also very happy .

Q6: What color? -> Q7: What were they doing?

Billy went to the farm to buy some beef for his brother 's birthday . When he arrived there he saw that all six of the cows were sad and had brown spots . The cows were all eating their breakfast in a big grassy meadow . He thought that the spots looked very strange so he went closer to the cows to get a better look . When he got closer he also saw that there were five white chickens sitting on the fence . The fence was painted blue and had some dirty black spots on it . Billy wondered where the dirty spots had come . Soon he got close to the chickens and they got scared . All five chickens flew away and went to eat some food . After Billy got a good look at the cows he went to the farmer to buy some beef . The farmer gave him four pounds of beef for ten dollars . Billy thought that it was a good deal so he went home and cooked his brother dinner . His brother was very happy with the dinner . Billy 's mom was also very happy .

Q7: What were they doing? -> Q8: Where?

Billy went to the farm to buy some beef for his brother 's birthday . When he arrived there he saw that all six of the cows were sad and had brown spots . The cows were all eating their breakfast in a big grassy meadow . He thought that the spots looked very strange so he went closer to the cows to get a better look . When he got closer he also saw that there were five white chickens sitting on the fence . The fence was painted blue and had some dirty black spots on it . Billy wondered where the dirty spots had come . Soon he got close to the chickens and they got scared . All five chickens flew away and went to eat some food . After Billy got a good look at the cows he went to the farmer to buy some beef . The farmer gave him four pounds of beef for ten dollars . Billy thought that it was a good deal so he went home and cooked his brother dinner . His brother was very happy with the dinner . Billy 's mom was also very happy .

Figure 4. The highlighted context indicates the QA model's focus shifts between consecutive turns.