Commonsense for Generative Multi-Hop Question Answering Tasks

Lisa Bauer* **Yicheng Wang*** **Mohit Bansal**
UNC Chapel Hill
{lbauer6, yicheng, mbansal}@cs.unc.edu

Abstract

Reading comprehension QA tasks have seen a recent surge in popularity, yet most works have focused on fact-finding extractive QA. We instead focus on a more challenging multi-hop generative task (NarrativeQA), which requires the model to reason, gather, and synthesize disjoint pieces of information within the context to generate an answer. This type of multi-step reasoning also often requires understanding implicit relations, which humans resolve via external, background commonsense knowledge. We first present a strong generative baseline that uses a multi-attention mechanism to perform multiple hops of reasoning and a pointer-generator decoder to synthesize the answer. This model performs substantially better than previous generative models, and is competitive with current state-of-the-art span prediction models. We next introduce a novel system for selecting grounded multi-hop relational commonsense information from ConceptNet via a pointwise mutual information and term-frequency based scoring function. Finally, we effectively use this extracted commonsense information to fill in gaps of reasoning between context hops, using a selectively-gated attention mechanism. This boosts the model's performance significantly (also verified via human evaluation), establishing a new state-of-the-art for the task. We also show that our background knowledge enhancements are generalizable and improve performance on QAngaroo-WikiHop, another multi-hop reasoning dataset.

1 Introduction

In this paper, we explore the task of machine reading comprehension (MRC) based QA. This task tests a model's natural language understanding capabilities by asking it to answer a question based on a passage of relevant content. Much progress has been made in reasoning-based MRC-QA on the bAbI dataset (Weston et al., 2016), which contains questions that require the combination of multiple disjoint pieces of evidence in the context. However, due to its synthetic nature, bAbI evidences have smaller lexicons and simpler passage structures when compared to human-generated text.

There also have been several attempts at the MRC-QA task on human-generated text. Large scale datasets such as CNN/DM (Hermann et al., 2015) and SQuAD (Rajpurkar et al., 2016) have made the training of end-to-end neural models possible. However, these datasets are fact-based and do not place heavy emphasis on multi-hop reasoning capabilities. More recent datasets such as QAngaroo (Welbl et al., 2018) have prompted a strong focus on multi-hop reasoning in very long texts. However, QAngaroo is an extractive dataset where answers are guaranteed to be spans within the context; hence, this is more focused on fact finding and linking, and does not require models to synthesize and generate new information.

We focus on the recently published NarrativeQA generative dataset (Kočiskỳ et al., 2018) that contains questions requiring multi-hop reasoning for long, complex stories and other narratives, which requires the model to go beyond fact linking and to synthesize non-span answers. Hence, models that perform well on previous reasoning tasks (Dhingra et al., 2018) have had limited success on this dataset. In this paper, we first propose the Multi-Hop Pointer-Generator Model (MHPGM), a strong baseline model that uses multiple hops of bidirectional attention, self-attention, and a pointer-generator decoder to effectively read and reason within a long passage and synthesize a coherent response. Our model achieves 41.49 Rouge-L and 17.33 METEOR on the summary

∗ Equal contribution (published at EMNLP 2018).
We publicly release all our code, models, and data at: https://github.com/yicheng-w/CommonSenseMultiHopQA

Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pages 4220–4230
Brussels, Belgium, October 31 - November 4, 2018.

subtask of NarrativeQA, substantially better than the performance of previous generative models.

Next, to address the issue that understanding human-generated text and performing long-distance reasoning on it often involves intermittent access to missing hops of external commonsense (background) knowledge, we present an algorithm for selecting useful, grounded multi-hop relational knowledge paths from ConceptNet (Speer and Havasi, 2012) via a pointwise mutual information (PMI) and term-frequency-based scoring function. We then present a novel method of inserting these selected commonsense paths between the hops of document-context reasoning within our model, via the Necessary and Optional Information Cell (NOIC), which employs a selectively-gated attention mechanism that utilizes commonsense information to effectively fill in gaps of inference. With these additions, we further improve performance on the NarrativeQA dataset, achieving 44.16 Rouge-L and 19.03 METEOR (also verified via human evaluation). We also provide manual analysis on the effectiveness of our commonsense selection algorithm.

Finally, to show the effectiveness and generalizability of our multi-hop reasoning and commonsense methods, we also tested our MHPGM and MHPGM+NOIC models on QAngaroo-WikiHop (Welbl et al., 2018), which is an extractive dataset for multi-hop reasoning on human-generated documents. We found that our background commonsense knowledge enhanced model achieved 1.5% higher accuracy than our strong baseline.

2 Related Work

Machine Reading Comprehension: MRC has long been a task used to assess a model's ability to understand and reason about language. Large scale datasets such as CNN/Daily Mail (Hermann et al., 2015) and SQuAD (Rajpurkar et al., 2016) have encouraged the development of many advanced, high performing attention-based neural models (Seo et al., 2017; Dhingra et al., 2017). Concurrently, datasets such as bAbI (Weston et al., 2016) have focused specifically on multi-step reasoning by requiring the model to reason with disjoint pieces of information. On this task, it has been shown that iteratively updating the query representation with information from the context can effectively emulate multi-step reasoning (Sukhbaatar et al., 2015).

More recently, there has been an increase in multi-paragraph, multi-hop inference QA datasets such as QAngaroo (Welbl et al., 2018) and NarrativeQA (Kočiskỳ et al., 2018). These datasets have much longer contexts than previous datasets, and answering a question often requires the synthesis of multiple discontiguous pieces of evidence. It has been shown that models designed for previous tasks (Seo et al., 2017; Kadlec et al., 2016) have limited success on these new datasets. In our work, we expand upon Gated Attention Network (Dhingra et al., 2017) to create a baseline model better suited for complex MRC datasets such as NarrativeQA by improving its attention and gating mechanisms, expanding its generation capabilities, and allowing access to external commonsense for connecting implicit relations.

Commonsense/Background Knowledge: Commonsense or background knowledge has been used for several tasks including opinion mining (Cambria et al., 2010), sentiment analysis (Poria et al., 2015, 2016), handwritten text recognition (Wang et al., 2013), and more recently, dialogue (Young et al., 2018; Ghazvininejad et al., 2018). These approaches add commonsense knowledge as relation triples or features from external databases. Recently, large-scale graphical commonsense databases such as ConceptNet (Speer and Havasi, 2012) use graphical structure to express intricate relations between concepts, but effective goal-oriented graph traversal has not been extensively used in previous commonsense incorporation efforts. Knowledge-base QA is a task in which systems are asked to find answers to questions by traversing knowledge graphs (Bollacker et al., 2008). Knowledge path extraction has been shown to be effective at the task (Bordes et al., 2014; Bao et al., 2016). We apply these techniques to MRC-QA by using them to extract useful commonsense knowledge paths that fully utilize the graphical nature of databases such as ConceptNet (Speer and Havasi, 2012).

Incorporation of External Knowledge: There have been several attempts at using external knowledge to boost model performance on a variety of tasks: Chen et al. (2018) showed that adding lexical information from semantic databases such as WordNet improves performance on NLI; Xu et al. (2017) used a gated recall-LSTM mechanism to incorporate commonsense information into to-

ken representations in dialogue.

In MRC, Weissenborn et al. (2017) integrated external background knowledge into an NLU model by using contextually-refined word embeddings which integrated information from ConceptNet (single-hop relations mapped to unstructured text) via a single layer bidirectional LSTM. Concurrently to our work, Mihaylov and Frank (2018) showed improvements on a cloze-style task by incorporating commonsense knowledge via a context-to-commonsense attention, where commonsense relations were extracted as triples. This work represented commonsense relations as key-value pairs and combined context representation and commonsense via a static gate.

Differing from previous works, we employ multi-hop commonsense paths (multiple connected edges within ConceptNet graph that give us information beyond a single relationship triple) to help with our MRC model. Moreover, we use this in tandem with our multi-hop reasoning architecture to incorporate different aspects of the commonsense relationship path at each hop, in order to bridge different inference gaps in the multi-hop QA task. Additionally, our model performs synthesis with its external, background knowledge as it generates, rather than extracts, its answer.

3 Methods

3.1 Multi-Hop Pointer-Generator Baseline

We first rigorously state the problem of generative QA as follows: given two sequences of input tokens: the context, $X^C = \{w_1^C, w_2^C, \ldots, w_n^C\}$ and the query, $X^Q = \{w_1^Q, w_2^Q, \ldots, w_m^Q\}$, the system should generate a series of answer tokens $X^a = \{w_1^a, w_2^a, \ldots, w_p^a\}$. As outlined in previous sections, an effective generative QA model needs to be able to perform several hops of reasoning over long and complex passages. It would also need to be able to generate coherent statements to answer complex questions while having the ability to copy rare words such as specific entities from the reading context. With these in mind, we propose the Multi-Hop Pointer-Generator Model (MHPGM) baseline, a novel combination of previous works with the following major components:

- **Embedding Layer**: The tokens are embedded into both learned word embeddings and pre-trained context-aware embeddings (ELMo (Peters et al., 2018)).
- **Reasoning Layer**: The embedded context is then passed through k reasoning cells, each of which iteratively updates the context representation with information from the query via BiDAF attention (Seo et al., 2017), emulating a single reasoning step within the multi-step reasoning process.
- **Self-Attention Layer**: The context representation is passed through a layer of self-attention (Cheng et al., 2016) to resolve long-term dependencies and co-reference within the context.
- **Pointer-Generator Decoding Layer**: A attention-pointer-generator decoder (See et al., 2017) that attends on and potentially copies from the context is used to create the answer.

The overall model is illustrated in Fig. 1, and the layers are described in further detail below.

Embedding layer: We embed each word from the context and question with a learned embedding space of dimension d. We also obtain context-aware embeddings for each word via the pre-trained embedding from language models (ELMo) (1024 dimensions). The embedded representation for each word in the context or question, $\mathbf{e}_i^C$ or $\mathbf{e}_i^Q \in \mathbb{R}^{d+1024}$, is the concatenation of its learned word embedding and ELMo embedding.

Reasoning layer: Our reasoning layer is composed of k reasoning cells (see Fig. 1), where each incrementally updates the context representation. The t^{th} reasoning cell's inputs are the previous step's output ($\{\mathbf{c}_i^{t-1}\}_{i=1}^n$) and the embedded question ($\{\mathbf{e}_i^Q\}_{i=1}^m$). It first creates step-specific context and query encodings via cell-specific bidirectional LSTMs:

$$\mathbf{u}^t = \text{BiLSTM}(\mathbf{c}^{t-1}); \qquad \mathbf{v}^t = \text{BiLSTM}(\mathbf{e}^Q)$$

Then, we use bidirectional attention (Seo et al., 2017) to emulate a hop of reasoning by focusing on relevant aspects of the context. Specifically, we first compute context-to-query attention:

$$S_{ij}^t = W_1^t \mathbf{u}_i^t + W_2^t \mathbf{v}_j^t + W_3^t (\mathbf{u}_i^t \odot \mathbf{v}_j^t)$$

$$p_{ij}^t = \frac{\exp(S_{ij}^t)}{\sum_{k=1}^m \exp(S_{ik}^t)}$$

$$(\mathbf{c_q})_i^t = \sum_{j=1}^m p_{ij}^t \mathbf{v}_j^t$$

where W_1^t, W_2^t, W_3^t are trainable parameters, and $\odot$ is elementwise multiplication. We then compute a query-to-context attention vector:

$$m_i^t = \max_{1 \le j \le m} S_{ij}^t$$

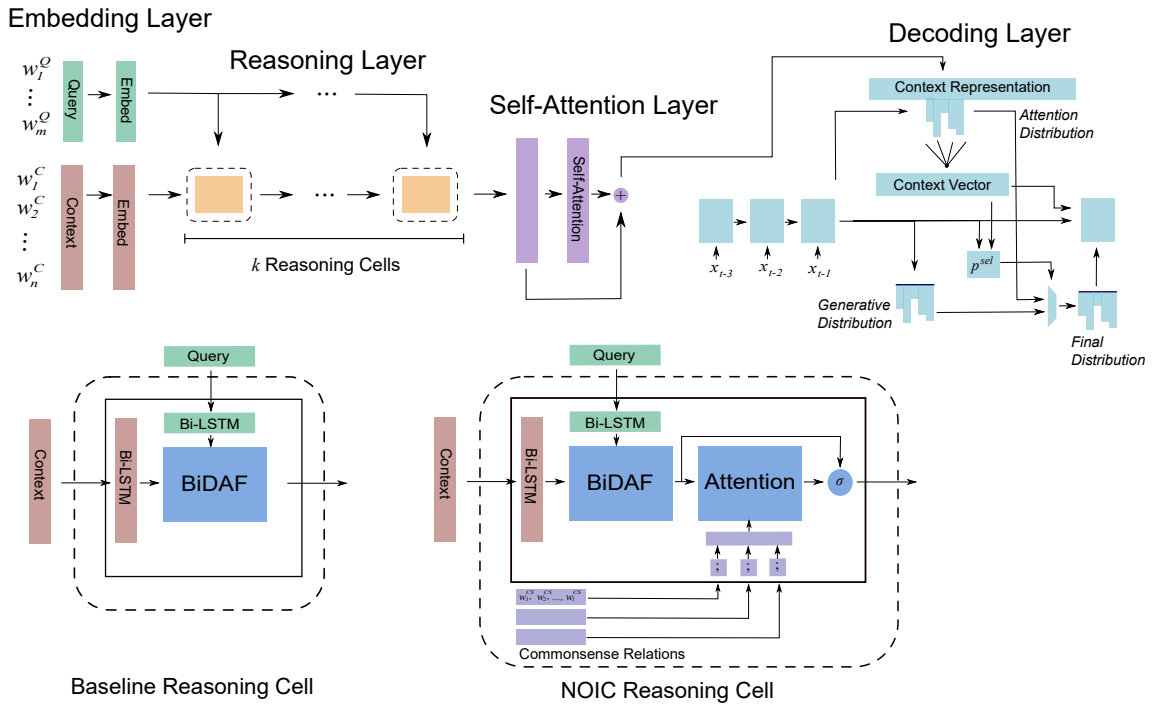

Figure 1: Architecture for our Multi-Hop Pointer-Generator Model, and the NOIC commonsense reasoning cell.

$$p_i^t = \frac{\exp(m_i^t)}{\sum_{j=1}^{n} \exp(m_j^t)}$$

$$\mathbf{q_c}^t = \sum_{i=1}^{n} p_i^t \mathbf{u}_i^t$$

We then obtain the updated context representation:

$$\mathbf{c}_i^t = [\mathbf{u}_i^t; (\mathbf{c_q})_i^t; \mathbf{u}_i^t \odot (\mathbf{c_q})_i^t; \mathbf{q_c}^t \odot (\mathbf{c_q})_i^t]$$

where ; is concatenation, $\mathbf{c}^t$ is the cell's output.

The initial input of the reasoning layer is the embedded context representation, i.e., $\mathbf{c}^0 = \mathbf{e}^C$, and the final output of the reasoning layer is the output of the last cell, $\mathbf{c}^k$.

Self-Attention Layer: As the final layer before answer generation, we utilize a residual static self-attention mechanism (Clark and Gardner, 2018) to help the model process long contexts with long-term dependencies. The input of this layer is the output of the last reasoning cell, $\mathbf{c}^k$. We first pass this representation through a fully-connected layer and then a bi-directional LSTM to obtain another representation of the context $\mathbf{c}^{SA}$. We obtain the self attention representation $\mathbf{c}'$:

$$S_{ij}^{SA} = W_4 \mathbf{c}_i^{SA} + W_5 \mathbf{c}_j^{SA} + W_6(\mathbf{c}_i^{SA} \odot \mathbf{c}_j^{SA})$$

$$p_{ij}^{SA} = \frac{\exp(S_{ij}^{SA})}{\sum_{k=1}^{n} \exp(S_{ik}^{SA})}$$

$$\mathbf{c'}_i = \sum_{j=1}^{n} p_{ij}^{SA} \mathbf{c}_j^{SA}$$

where W_4, W_5, and W_6 are trainable parameters.

The output of the self-attention layer is generated by another layer of bidirectional LSTM.

$$\mathbf{c}'' = \text{BiLSTM}([\mathbf{c}'; \mathbf{c}^{SA}; \mathbf{c}' \odot \mathbf{c}^{SA}]$$

Finally, we add this residually to $\mathbf{c}^k$ to obtain the encoded context $\mathbf{c} = \mathbf{c}^k + \mathbf{c}''$.

Pointer-Generator Decoding Layer: Similar to the work of See et al. (2017), we use a pointer-generator model attending on (and potentially copying from) the context.

At decoding step t, the decoder receives the input $\mathbf{x}_t$ (embedded representation of last timestep's output), the last time step's hidden state $\mathbf{s}_{t-1}$ and context vector $\mathbf{a}_{t-1}$. The decoder computes the current hidden state $\mathbf{s}_t$ as:

$$\mathbf{s}_t = \text{LSTM}([\mathbf{x}_t; \mathbf{a}_{t-1}], \mathbf{s}_{t-1})$$

This hidden state is then used to compute a probability distribution over the generative vocabulary:

$$P_{gen} = \text{softmax}(W_{gen}\mathbf{s}_t + \mathbf{b}_{gen})$$

We employ Bahdanau attention mechanism (Bahdanau et al., 2015) to attend over the context ($\mathbf{c}$ being the output of self-attention layer):

$$\alpha_i = \mathbf{v}^\intercal \tanh(W_c \mathbf{c}_i + W_s \mathbf{s}_t + b_{attn})$$

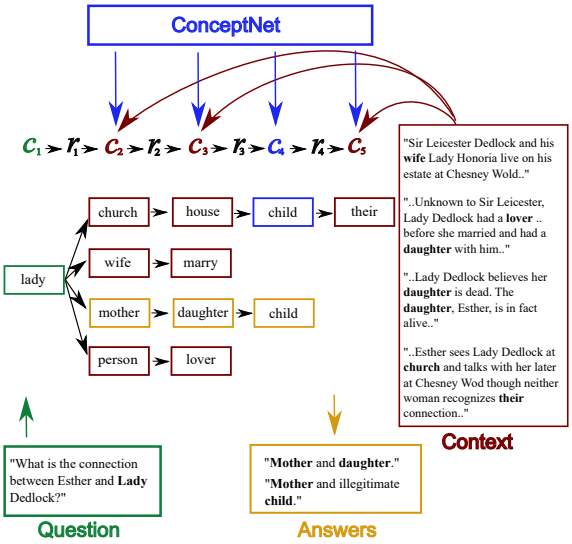

Figure 2: Commonsense selection approach.

$$\hat{\alpha}_i = \frac{\exp(\alpha_i)}{\sum_{j=1}^{n} \exp(\alpha_j)}$$

$$\mathbf{a}_t = \sum_{i=1}^{n} \hat{\alpha}_i \mathbf{c}_i$$

We utilize a pointer mechanism that allows the decoder to directly copy tokens from the context based on $\hat{\alpha}_i$. We calculate a selection distribution $\mathbf{p}^{sel} \in \mathbb{R}^2$, where p_1^{sel} is the probability of generating a token from P_{gen} and p_2^{sel} is the probability of copying a word from the context:

$$\mathbf{o} = \sigma(W_a \mathbf{a}_t + W_x \mathbf{x}_t + W_s \mathbf{s}_t + b_{ptr})$$

$$\mathbf{p}^{sel} = \text{softmax}(\mathbf{o})$$

Our final output distribution at timestep t is a weighted sum of the generative distribution and the copy distribution:

$$P_t(w) = p_1^{sel} P_{gen}(w) + p_2^{sel} \sum_{i:w_i^C = w} \hat{\alpha}_i$$

3.2 Commonsense Selection and Representation

In QA tasks that require multiple hops of reasoning, the model often needs knowledge of relations not directly stated in the context to reach the correct conclusion. In the datasets we consider, manual analysis shows that external knowledge is frequently needed for inference (see Table 1).

Even with a large amount of training data, it is very unlikely that a model is able to learn every nuanced relation between concepts and apply the correct ones (as in Fig. 2) when reasoning about a question. We remedy this issue by introducing *grounded* commonsense (background) information using relations between concepts from ConceptNet (Speer and Havasi, 2012)[1] that help inference by introducing useful connections between concepts in the context and question.

Dataset	Outside Knowledge Required
WikiHop	11%
NarrativeQA	42%

Table 1: Qualitative analysis of commonsense requirements. WikiHop results are from Welbl et al. (2018); NarrativeQA results are from our manual analysis (on the validation set).

Due to the size of the semantic network and the large amount of unnecessary information, we need an effective way of selecting relations which provides novel information while being grounded by the context-query pair. Our commonsense selection strategy is twofold: (1) collect potentially relevant concepts via a tree construction method aimed at selecting with high recall candidate reasoning paths, and (2) rank and filter these paths to ensure both the quality and variety of added information via a 3-step scoring strategy (initial node scoring, cumulative node scoring, and path selection). We will refer to Fig. 2 as a running example throughout this section.[2]

3.2.1 Tree Construction

Given context C and question Q, we want to construct paths grounded in the pair that emulate reasoning steps required to answer the question. In this section, we build 'prototype' paths by constructing trees rooted in concepts in the query with the following branching steps[3] to emulate multi-hop reasoning process. For each concept c_1 in the question, we do:

Direct Interaction: In the first level, we select relations r_1 from ConceptNet that directly link c_1 to a concept within the context, $c_2 \in C$, e.g., in Fig. 2, we have *lady* $\rightarrow$ *church*, *lady* $\rightarrow$ *mother*, *lady* $\rightarrow$ *person*.

Multi-Hop: We then select relations in ConceptNet r_2 that link c_2 to another concept in the context, $c_3 \in C$. This emulates a potential reason-

[1] A semantic network where the nodes are individual concepts (words or phrases) and the edges describe directed relations between them (e.g., ⟨island, UsedFor, vacation⟩).

[2] We release all our commonsense extraction code and the extracted commonsense data at: https://github.com/yicheng-w/CommonSenseMultiHopQA

[3] If we are unable to find a relation that satisfies the condition, we keep the steps up to and including the node.

ing hop within the context of the MRC task, e.g., *church* → *house*, *mother* → *daughter*, *person* → *lover*.

Outside Knowledge: We then allow an unconstrained hop into c_3's neighbors in ConceptNet, getting to $c_4 \in \text{nbh}(c_3)$ via r_3 ($\text{nbh}(v)$ is the set of nodes that can be reached from v in one hop). This emulates the gathering of useful external information to complete paths within the context, e.g., *house* → *child*, *daughter* → *child*.

Context-Grounding: To ensure that the external knowledge is indeed helpful to the task, and also to explicitly link 2nd degree neighbor concepts within the context, we finish the process by grounding it again into context by connecting c_4 to $c_5 \in C$ via r_4, e.g., *child* → *their*.

3.2.2 Rank and Filter

This tree building process collects a large number of potentially relevant and useful paths. However, this step also introduces a large amount of noise. For example, given the question and full context (not depicted in the figure) in Fig. 2, we obtain the path "*between* → *hard* → *being* → *cottage* → *country*" using our tree building method, which is not relevant to our question. Therefore, to improve the precision of useful concepts, we rank these knowledge paths by their relevance and filter out noise using the following 3-step scoring method:

Initial Node Scoring: We want to select paths with nodes that are important to the context, in order to provide the most useful commonsense relations. We approximate importance and saliency for concepts in the context by their term-frequency, under the heuristic that important concepts occur more frequently. Thus we score $c \in \{c_2, c_3, c_5\}$ by: $\text{score}(c) = \text{count}(c)/|C|$, where $|C|$ is the context length and count() is the number of times a concept appears in the context. In Fig. 2, this ensures that concepts like *daughter* are scored highly due to their frequency in the context.

For c_4, we use a special scoring function as it is an unconstrained hop into ConceptNet. We want c_4 to be a logically consistent next step in reasoning following the path of c_1 to c_3, e.g., in Fig. 2, we see that *child* is a logically consistent next step after the partial path of *mother* → *daughter*. We approximate this based on the heuristic that logically consistent paths occur more frequently. Therefore, we score this node via Pointwise Mutual Information (PMI) between the partial path c_{1-3} and c_4: $\text{PMI}(c_4, c_{1-3}) = \log(\mathbb{P}(c_4, c_{1-3})/\mathbb{P}(c_4)\mathbb{P}(c_{1-3}))$, where

$$\mathbb{P}(c_4, c_{1-3}) = \frac{\text{\# of paths connecting } c_1, c_2, c_3, c_4}{\text{\# of distinct paths of length 4}}$$

$$\mathbb{P}(c_4) = \frac{\text{\# of nodes that can reach } c_4}{|\text{ConceptNet}|}$$

$$\mathbb{P}(c_{1-3}) = \frac{\text{\# of paths connecting } c_1, c_2, c_3}{\text{\# of paths of length 3}}$$

Further, it is well known that PMI has high sensitivity to low-frequency values, thus we use normalized PMI (NPMI) (Bouma, 2009): $\text{score}(c_4) = \text{PMI}(c_4, c_{1-3})/(-\log \mathbb{P}(c_4, c_{1-3}))$.

Since the branching at each juncture represents a hop in the multi-hop reasoning process, and hops at different levels or with different parent nodes do not 'compete' with each other, we normalize each node's score against its siblings:

$$\text{n-score}(c) = \text{softmax}_{\text{siblings}(c)}(\text{score}(c)).$$

Cumulative Node Scoring: We want to add commonsense paths consisting of multiple hops of relevant information, thus we re-score each node based not only on its relevance and saliency but also that of its tree descendants.

We do this by computing a cumulative node score from the bottom up, where at the leaf nodes, we have c-score = n-score, and for c_l not a leaf node, we have $\text{c-score}(c_l) = \text{n-score}(c_l) + f(c_l)$ where f of a node is the average of the c-scores of its top 2 highest scoring children.

For example, given the paths *lady* → *mother* → *daughter*, *lady* → *mother* → *married*, and *lady* → *mother* → *book*, we start the cumulative scoring at the leaf nodes, which in this case are *daughter*, *married*, and *book*, where *daughter* and *married* are scored much higher than *book* due to their more frequent occurrences. Then, to cumulatively score *mother*, we would take the average score of its two highest scoring children (in this case *married* and *daughter*) and compound that with the score of *mother* itself. Note that the poor scoring of the irrelevant concept *book* does not affect the scoring of *mother*, which is quite high due to the concept's frequent occurrence and the relevance of its top scoring children.

Path Selection: We select paths in a top-down breath-first fashion in order to add information relevant to different parts of the context. Starting at the root, we recursively take two of its children with the highest cumulative scores until we reach a leaf, selecting up to $2^4 = 16$ paths. For example,

if we were at node *mother*, this allows us to select the child node *daughter* and *married* over the child node *book*. These selected paths, as well as their partial sub-paths, are what we add as external information to the QA model, i.e., we add the complete path ⟨lady, AtLocation, church, RelatedTo, house, RelatedTo, child, RelatedTo, their⟩, but also truncated versions of the path, including ⟨lady, AtLocation, church, RelatedTo, house, RelatedTo, child⟩. We directly give these paths to the model as sequences of tokens.[4]

Overall, our sampling strategy provides the knowledge that a *lady* can be a *mother* and that *mother* is connected to *daughter*. This creates a logical connection between *lady* and *daughter* which helps highlight the importance of our second piece of evidence (see Fig. 2). Likewise, the commonsense information we extracted create a similar connection in our third piece of evidence, which states the explicit connection between *daughter* and *Esther*. We also successfully extract a more story context-centric connection, in which commonsense provides the knowledge that a *lady* is at the location *church*, which directs to another piece of evidence in the context. Additionally, this path also encodes a relation between *lady* and *child*, by way of *church*, which is how *lady* and *Esther* are explicitly connected in the story.

3.3 Commonsense Model Incorporation

Given the list of commonsense logic paths as sequences of words: $X^{CS} = \{w_1^{CS}, w_2^{CS}, \ldots, w_l^{CS}\}$ where w_i^{CS} represents the list of tokens that make up a single path, we first embed these commonsense tokens into the learned embedding space used by the model, giving us the embedded commonsense tokens, $\mathbf{e}_{ij}^{CS} \in \mathbb{R}^d$. We want to use these commonsense paths to fill in the gaps of reasoning between hops of inference. Thus, we propose Necessary and Optional Information Cell (NOIC), a variation of our base reasoning cell used in the reasoning layer that is capable of incorporating optional helpful information.

NOIC This cell is an extension to the base reasoning cell that allows the model to use commonsense information to fill in gaps of reasoning. An example of this is on the bottom left of Fig. 1, where we see that the cell first performs the operations done in the base reasoning cell and then adds optional, commonsense information.

At reasoning step t, after obtaining the output of the base reasoning cell, $\mathbf{c}^t$, we create a cell-specific representation for commonsense information by concatenating the embedded commonsense paths so that each path has a single vector representation, $\mathbf{u}_i^{CS}$. We then project it to the same dimension as $\mathbf{c}_i^t$: $\mathbf{v}_i^{CS} = \text{ReLU}(W\mathbf{u}_i^{CS} + b)$ where W and b are trainable parameters.

We use an attention layer to model the interaction between commonsense and the context:

$$S_{ij}^{CS} = W_1^{CS}\mathbf{c}_i^t + W_2^{CS}\mathbf{v}_j^{CS} + W_3^{CS}(\mathbf{c}_i^t \odot \mathbf{v}_j^{CS})$$

$$p_{ij}^{CS} = \frac{\exp(S_{ij}^{CS})}{\sum_{k=1}^{l}\exp(S_{ij}^{CS})}$$

$$\mathbf{c}_i^{CS} = \sum_{j=1}^{l} p_{ij}^{CS}\mathbf{v}_j^{CS}$$

Finally, we combine this commonsense-aware context representation with the original $\mathbf{c}_i^t$ via a sigmoid gate, since commonsense information is often not necessary at every step of inference:

$$\mathbf{z}_i = \sigma(W_z[\mathbf{c}_i^{CS}; \mathbf{c}_i^t] + b_z)$$

$$(\mathbf{c_o})_i^t = \mathbf{z}_i \odot \mathbf{c}_i^t + (1 - \mathbf{z}_i) \odot \mathbf{c}_i^{CS}$$

We use $\mathbf{c_o}^t$ as the output of the current reasoning step instead of $\mathbf{c}^t$. As we replace each base reasoning cell with NOIC, we selectively incorporate commonsense at every step of inference.

4 Experimental Setup

Datasets: We report results on two multi-hop reasoning datasets: generative NarrativeQA (Kočiskỳ et al., 2018) (summary subtask) and extractive QAngaroo WikiHop (Welbl et al., 2018). For multiple-choice WikiHop, we rank candidate responses by their generation probability. Similar to previous works (Dhingra et al., 2018), we use the non-oracle, unmasked and not-validated dataset.

Evaluation Metrics: We evaluate NarrativeQA on the metrics proposed by its original authors: Bleu-1, Bleu-4 (Papineni et al., 2002), METEOR (Banerjee and Lavie, 2005) and Rouge-L (Lin, 2004). We also evaluate on CIDEr (Vedantam et al., 2015) which emphasizes annotator consensus. For WikiHop, we evaluate on accuracy.[5]

More dataset, metric, and all other training details are in the supplementary.

[4] In cases where more than one relation can be used to make a hop, we pick one at random.

[5] Due to the 2-week evaluation wait-time on the non-public test set, we instead train our model on a sub-section of the training set, pick hyperparameters based on a small

Model	BLEU-1	BLEU-4	METEOR	Rouge-L	CIDEr
Seq2Seq (Kočiskỳ et al., 2018)	15.89	1.26	4.08	13.15	-
ASR (Kočiskỳ et al., 2018)	23.20	6.39	7.77	22.26	-
BiDAF† (Kočiskỳ et al., 2018)	33.72	15.53	15.38	36.30	-
BiAttn + MRU-LSTM† (Tay et al., 2018)	36.55	19.79	17.87	41.44	-
MHPGM	40.24	17.40	17.33	41.49	139.23
MHPGM+ NOIC	**43.63**	**21.07**	**19.03**	**44.16**	**152.98**

Table 2: Results across different metrics on the test set of NarrativeQA-summaries task. † indicates span prediction models trained on the Rouge-L retrieval oracle.

Model	Acc (%)
BiDAF (Welbl et al., 2018)	42.09
Coref-GRU (Dhingra et al., 2018)	56.00
MHPGM	56.74
MHPGM+ NOIC	**58.22**

Table 3: Results of our models on WikiHop dataset.

5 Results

5.1 Main Experiment

The results of our model on both NarrativeQA and WikiHop with and without commonsense incorporation are shown in Table 2 and Table 3. We see empirically that our model outperforms all generative models on NarrativeQA, and is competitive with the top span prediction models. Furthermore, with the NOIC commonsense integration, we were able to further improve performance ($p < 0.001$ on all metrics[6]), establishing a new state-of-the-art for the task. We also see that our model performs well on WikiHop,[7] and is further improved via the addition of commonsense ($p < 0.001$), demonstrating the generalizability of both our model and commonsense addition techniques.[8]

5.2 Model Ablations

We also tested the effectiveness of each component of our architecture as well as the effectiveness of adding commonsense information on the NarrativeQA validation set, with results shown in Table 4. Experiment 1 and 5 are our models presented in Table 2. Experiment 2 demonstrates the importance of multi-hop attention by showing that if we only allow one hop of attention (even with all other components of the model, including ELMo embeddings) the model's performance decreases by over 12 Rouge-L points. Experiment 3 and 4 demonstrate the effectiveness of other parts of our model. We see that ELMo embeddings (Peters et al., 2018) were also important for the model's performance and that self-attention is able to contribute significantly to performance on top of other components of the model. Finally, we see that effectively introducing external knowledge via our commonsense selection algorithm and NOIC can improve performance even further on top of our strong baseline.

5.3 Commonsense Ablations

We also conducted experiments testing the effectiveness of our commonsense selection and incorporation techniques. We first tried to naively add ConceptNet information by initializing the word embeddings with the ConceptNet-trained embeddings, NumberBatch (Speer and Havasi, 2012) (we also change embedding size from 256 to 300). Then, to verify the effectiveness of our commonsense selection and grounding algorithm, we test our best model on in-domain noise by giving each context-query pair a set of random relations grounded in other context-query pairs. This should teach the model about general commonsense relations present in the domain of NarrativeQA but does not provide grounding that fills in specific hops of inference. We also experimented with a simpler commonsense extraction method of using a single hop from the query to the context. The results of these are shown in Table 5, where we see that neither NumberBatch nor random-relationships nor single-hop commonsense offer statistically significant improvements[9],

(500 examples) held-out part of the training set, and test on the original validation set (by treating it as an unseen test set). We will promptly include the non-public test set results in the next version and at: https://github.com/yicheng-w/CommonSenseMultiHopQA

[6]Stat. significance computed using bootstrap test with 100K iterations (Noreen, 1989; Efron and Tibshirani, 1994).

[7]Note that we compare our model's performance to other models' tuned performance on the development set and ours is still equal or better.

[8]All results here are for the standard (non-oracle) unmasked and not-validated dataset. Welbl et al. (2018) has reported higher numbers on different data settings which are not comparable to our results.

[9]The improvement in Rouge-L and METEOR for all three ablation approaches have $p \geq 0.15$ with the bootstrap test.

#	Ablation	B-1	B-4	M	R	C
1	-	42.3	18.9	18.3	44.9	151.6
2	$k = 1$	32.5	11.7	12.9	32.4	95.7
3	- ELMo	32.8	12.7	13.6	33.7	103.1
4	- Self-Attn	37.0	16.4	15.6	38.6	125.6
5	+ NOIC	**46.0**	**21.9**	**20.7**	**48.0**	**166.6**

Table 4: Model ablations on NarrativeQA val-set.

Commonsense	B-1	B-4	M	R	C
None	42.3	18.9	18.3	44.9	151.6
NumberBatch	42.6	19.6	18.6	44.4	148.1
Random Rel.	43.3	19.3	18.6	45.2	151.2
Single Hop	42.1	19.9	18.2	44.0	148.6
Grounded Rel.	**45.9**	**21.9**	**20.7**	**48.0**	**166.6**

Table 5: Commonsense ablations on NarrativeQA val-set.

whereas our commonsense selection and incorporation mechanism improves performance significantly across all metrics. We also present several examples of extracted commonsense and its model attention visualization in the supplementary.

6 Human Evaluation Analysis

We also conduct human evaluation analysis on both the quality of the selected commonsense relations, as well as the performance of our final model.

Commonsense Selection: We conducted manual analysis on a 50 sample subset of the NarrativeQA test set to check the effectiveness of our commonsense selection algorithm. Specifically, given a context-query pair, as well as the commonsense selected by our algorithm, we conduct two independent evaluations: (1) was any external commonsense knowledge necessary for answering the question?; (2) were the commonsense relations provided by our algorithm relevant to the question? The result for these two evaluations as well as how they overlap with each other are shown in Table 6, where we see that 50% of the cases required external commonsense knowledge, and on a majority (34%) of those cases our algorithm was able to select the correct/relevant commonsense information to fill in gaps of inference. We also see that in general, our algorithm was able to provide useful commonsense 48% of the time.

Model Performance: We also conduct human evaluation to verify that our commonsense incorporated model was indeed better than MHPGM. We randomly selected 100 examples from the NarrativeQA test set, along with both models' predicted answers, and for each datapoint, we asked

	Commonsense Required	
	Yes	No
Relevant CS Extracted	34%	14%
Irrelevant CS Extracted	16%	36%

Table 6: NarrativeQA's commonsense requirements and effectiveness of commonsense selection algorithm.

MHPGM+NOIC better	23%
MHPGM better	15%
Indistinguishable (Both-good)	41%
Indistinguishable (Both-bad)	21%

Table 7: Human evaluation on the output quality of the MHPGM+NOIC vs. MHPGM in terms of correctness.

3 external human evaluators (fluent English speakers) to decide (without knowing which model produced each response) if one is strictly better than the other, or that they were similar in quality (both-good or both-bad). As shown in Table 7, we see that the human evaluation results are in agreement with that of the automatic evaluation metrics: our commonsense incorporation has a reasonable impact on the overall correctness of the model. The inter-annotator agreement had a Fleiss $\kappa = 0.831$, indicating 'almost-perfect' agreement between the annotators (Landis and Koch, 1977).

7 Conclusion

We present an effective reasoning-generative QA architecture that is a novel combination of previous work, which uses multiple hops of bidirectional attention and a pointer-generator decoder to effectively perform multi-hop reasoning and synthesize a coherent and correct answer. Further, we introduce an algorithm to select grounded, useful paths of commonsense knowledge to fill in the gaps of inference required for QA, as well a Necessary and Optional Information Cell (NOIC) which successfully incorporates this information during multi-hop reasoning to achieve the new state-of-the-art on NarrativeQA.

Acknowledgments

We thank the reviewers for their helpful comments. This work was supported by DARPA (YFA17-D17AP00022), Google Faculty Research Award, Bloomberg Data Science Research Grant, and NVidia GPU awards. The views contained in this article are those of the authors and not of the funding agency.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *ICLR*.

Satanjeev Banerjee and Alon Lavie. 2005. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pages 65–72.

Junwei Bao, Nan Duan, Zhao Yan, Ming Zhou, and Tiejun Zhao. 2016. Constraint-based question answering with knowledge graph. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 2503–2514.

Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 1247–1250. AcM.

Antoine Bordes, Sumit Chopra, and Jason Weston. 2014. Question answering with subgraph embeddings. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 615–620.

Gerlof Bouma. 2009. Normalized (pointwise) mutual information in collocation extraction. *Proceedings of GSCL*, pages 31–40.

Erik Cambria, Amir Hussain, Tariq Durrani, Catherine Havasi, Chris Eckl, and James Munro. 2010. Sentic computing for patient centered applications. In *Signal Processing (ICSP), 2010 IEEE 10th International Conference on*, pages 1279–1282. IEEE.

Qian Chen, Xiaodan Zhu, Zhen-Hua Ling, Diana Inkpen, and Si Wei. 2018. Neural natural language inference models enhanced with external knowledge. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2406–2417. Association for Computational Linguistics.

Jianpeng Cheng, Li Dong, and Mirella Lapata. 2016. Long short-term memory-networks for machine reading. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 551–561.

Christopher Clark and Matt Gardner. 2018. Simple and effective multi-paragraph reading comprehension. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Long Papers)*, pages 845–855.

Bhuwan Dhingra, Qiao Jin, Zhilin Yang, William Cohen, and Ruslan Salakhutdinov. 2018. Neural models for reasoning over multiple mentions using coreference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, volume 2, pages 42–48.

Bhuwan Dhingra, Hanxiao Liu, Zhilin Yang, William Cohen, and Ruslan Salakhutdinov. 2017. Gated-attention readers for text comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1832–1846.

Bradley Efron and Robert J Tibshirani. 1994. *An introduction to the bootstrap*. CRC press.

Marjan Ghazvininejad, Chris Brockett, Ming-Wei Chang, Bill Dolan, Jianfeng Gao, Wen-tau Yih, and Michel Galley. 2018. A knowledge-grounded neural conversation model. In *AAAI*.

Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *Advances in Neural Information Processing Systems*, pages 1693–1701.

Rudolf Kadlec, Martin Schmid, Ondrej Bajgar, and Jan Kleindienst. 2016. Text understanding with the attention sum reader network. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pages 908–918.

Tomáš Kočiskỳ, Jonathan Schwarz, Phil Blunsom, Chris Dyer, Karl Moritz Hermann, Gáabor Melis, and Edward Grefenstette. 2018. The narrativeqa reading comprehension challenge. *Transactions of the Association of Computational Linguistics*, 6:317–328.

J Richard Landis and Gary G Koch. 1977. The measurement of observer agreement for categorical data. *biometrics*, pages 159–174.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. *Text Summarization Branches Out*.

Todor Mihaylov and Anette Frank. 2018. Knowledgeable reader: enhancing cloze-style reading comprehension with external commonsense knowledge. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 821–832. Association for Computational Linguistics.

Eric W Noreen. 1989. *Computer-intensive methods for testing hypotheses*. Wiley New York.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 311–318. Association for Computational Linguistics.

Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of NAACL*.

Soujanya Poria, Erik Cambria, Alexander Gelbukh, Federica Bisio, and Amir Hussain. 2015. Sentiment data flow analysis by means of dynamic linguistic patterns. *IEEE Computational Intelligence Magazine*, 10(4):26–36.

Soujanya Poria, Iti Chaturvedi, Erik Cambria, and Federica Bisio. 2016. Sentic LDA: Improving on LDA with semantic similarity for aspect-based sentiment analysis. In *Neural Networks (IJCNN), 2016 International Joint Conference on*, pages 4465–4473. IEEE.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392.

Abigail See, Peter J Liu, and Christopher D Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1073–1083.

Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. 2017. Bidirectional attention flow for machine comprehension. In *ICLR*.

Robert Speer and Catherine Havasi. 2012. Representing general relational knowledge in ConceptNet 5. In *LREC*, pages 3679–3686.

Sainbayar Sukhbaatar, Jason Weston, Rob Fergus, et al. 2015. End-to-end memory networks. In *Advances in Neural Information Processing Systems*, pages 2440–2448.

Yi Tay, Luu Anh Tuan, and Siu Cheung Hui. 2018. Multi-range reasoning for machine comprehension. *arXiv preprint arXiv:1803.09074*.

Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. 2015. Cider: Consensus-based image description evaluation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4566–4575.

Qiu-Feng Wang, Erik Cambria, Cheng-Lin Liu, and Amir Hussain. 2013. Common sense knowledge for handwritten chinese text recognition. *Cognitive Computation*, 5(2):234–242.

Dirk Weissenborn, Tomáš Kočiskỳ, and Chris Dyer. 2017. Dynamic integration of background knowledge in neural NLU systems. *arXiv preprint arXiv:1706.02596*.

Johannes Welbl, Pontus Stenetorp, and Sebastian Riedel. 2018. Constructing datasets for multi-hop reading comprehension across documents. *Transactions of the Association of Computational Linguistics*, 6:287–302.

Jason Weston, Antoine Bordes, Sumit Chopra, Alexander M Rush, Bart van Merriënboer, Armand Joulin, and Tomas Mikolov. 2016. Towards ai-complete question answering: A set of prerequisite toy tasks. In *ICLR*.

Zhen Xu, Bingquan Liu, Baoxun Wang, Chengjie Sun, and Xiaolong Wang. 2017. Incorporating loose-structured knowledge into LSTM with recall gate for conversation modeling. In *International Joint Conference on Neural Networks*.

Tom Young, Erik Cambria, Iti Chaturvedi, Minlie Huang, Hao Zhou, and Subham Biswas. 2018. Augmenting end-to-end dialog systems with commonsense knowledge. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence (AAAI-18)*.