HOP, UNION, GENERATE: Explainable Multi-hop Reasoning without Rationale Supervision

Wenting Zhao and **Justin T. Chiu** and **Claire Cardie** and **Alexander M. Rush**
Department of Computer Science
Cornell University
{wz346,jtc257,ctc9,arush}@cornell.edu

Abstract

Explainable multi-hop question answering (QA) not only predicts answers but also identifies rationales, i. e. subsets of input sentences used to derive the answers. This problem has been extensively studied under the supervised setting, where both answer and rationale annotations are given. Because rationale annotations are expensive to collect and not always available, recent efforts have been devoted to developing methods that do not rely on supervision for rationales. However, such methods have limited capacities in modeling interactions between sentences, let alone reasoning across multiple documents. This work proposes a principled, probabilistic approach for training explainable multi-hop QA systems without rationale supervision. Our approach performs multi-hop reasoning by explicitly modeling rationales as sets, enabling the model to capture interactions between documents and sentences within a document. Experimental results show that our approach is more accurate at selecting rationales than the previous methods, while maintaining similar accuracy in predicting answers.

1 Introduction

Multi-hop reasoning is an important capability for any intelligent machine comprehension system. Question answering (QA) is a common application for evaluating a system's ability to reason across multiple steps (Geva et al., 2021; Yang et al., 2018; Welbl et al., 2018). Large language models have achieved tremendous success on challenging QA tasks, even in the few-shot setting (Wei et al., 2022). However, Min et al. (2019) and Chen and Durrett (2019) demonstrate that these models, in reality, often bypass multi-hop reasoning by performing shallow pattern matching, resulting in poor generalization ability (Tang et al., 2021). To avoid predictions made from such reasoning shortcuts, it is important to understand the series of steps the systems follow to derive the answers.

This work explores the challenge of building explainable multi-hop QA systems, which, in addition to predicting an answer, also identify a *rationale* – the set of sentences that lead to the answer. Depending on task specifications, the rationale can be within a single document, or span across multiple documents. Explainable multi-hop QA has been extensively studied in the supervised setting, where both rationale annotations and answer annotations are given. These approaches either apply multi-task loss functions (Joshi et al., 2020; Groeneveld et al., 2020; DeYoung et al., 2020) or design specialized network architectures (Tu et al., 2019; Fang et al., 2020). However, having access to rationale annotations is a strong assumption. In practice, they are expensive to collect (Geva et al., 2021), less available than answer annotations (Welbl et al., 2018), and can suffer from low agreement rates between annotators (Zhang et al., 2020).

Researchers have thus explored approaches that do not require rationale annotations (Lewis et al., 2020b; Glockner et al., 2020; Atanasova et al., 2022). However, these previous approaches limit their reasoning to information from 1 or 2 sentences, and so they cannot be applied in *multi-hop* scenarios, i.e. QA tasks that require making connections between several pieces of information across sentences and across documents. Additionally, these methods are either restricted to only work for multiple-choice QA, or restricted to only produce rationales at the document level but not at the sentence level.

We propose HOP, UNION, GENERATE (HUG), a principled, probabilistic approach for training explainable multi-hop QA systems without rationale supervision. HUG overcomes the two-sentence limitation of previous methods by directly reasoning about rationales as *sets* of sentences, while also extending rationale prediction to the multi-document setting. We show an overview of HUG in Figure 1. HUG leverages the naturally hierarchi-

Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, pages 16119–16130

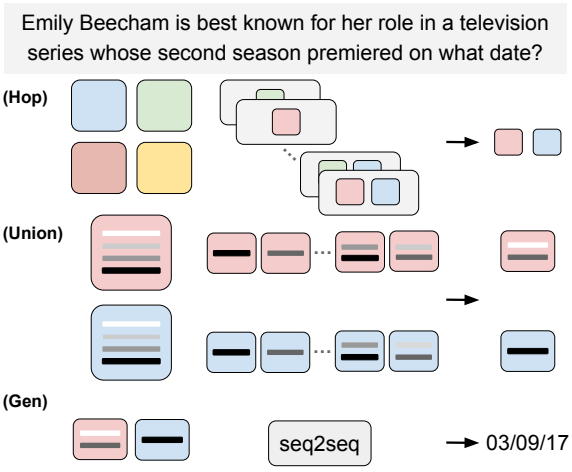

Figure 1: An overview of HUG, which proceeds in three stages. **Hop** explicitly considers all possible document sets and selects the most likely document set, **Union** explicitly considers all sentence subsets and chooses the most likely sentence subset within each selected document, and **Generate** combines the chosen sentence subsets and generates an answer.

cal structure of text and proceeds in three stages – it first selects the relevant set of documents given the question (Hop); then, it selects a subset of sentences within each of the relevant documents and collects them together (Union); finally, it generates an answer via a seq2seq model with all the collected sentences (Generate). The key to multi-hop reasoning in HUG is modeling each selection as an explicit distribution over sets. A probabilistic set distribution compares non-contiguous and variable size rationales, affording HUG flexibility for rationale selection.

Training a set-prediction model quickly becomes intractable as the size increases. We make two algorithmic choices that lead to tractable training for HUG. Treating rationales as a latent variable requires HUG to marginalize over all possible rationales, leading to an intractable learning objective. HUG overcomes this issue by performing sampling in a hierarchical way – it first identifies the most promising documents and then the most promising sentences within those documents. Second, multi-hop QA often involves reasoning over long documents, which is challenging due to the computational complexity of encoding long documents with neural models such as transformers. To make this encoding efficient, HUG performs computation in the embedding space.

We empirically evaluate HUG on three different multi-hop QA datasets: HotpotQA (Yang et al., 2018), MuSiQue (Trivedi et al., 2022), FEVER (DeYoung et al., 2020), and MultiRC (DeYoung et al., 2020). Results show that, for both selecting rationales and predicting answers, HUG is better than a number of state-of-the-art semi-supervised and unsupervised methods (Chen et al., 2019; Lewis et al., 2020b; Glockner et al., 2020; Atanasova et al., 2022) on all of the datasets. We also demonstrate that HUG combined with larger language models consistently produces better performance. We then analyze performance according to different types of multi-hop reasoning and show that by explicitly modeling multi-hop reasoning, HUG achieves a large improvement on the reasoning type that requires bridging entities.

2 Related Work

Explainable methods for multi-hop QA. Active research has been devoted to collecting human rationales for a wide range of QA tasks; a recent survey has identified 65 datasets that provide explanation annotations (Wiegreffe and Marasovic, 2021). The appearance of such datasets has enabled rapid progress in supervised methods for extracting reasoning chains; we refer readers to Thayaparan et al. (2020) for a comprehensive survey. While these supervised methods such as Qi et al. (2019) have achieved tremendous success on retrieving rationales, even in the open-domain setting [1], they can only be applied when rationale annotations are available. However, such annotations do not always exist – Welbl et al. (2018) and Yu et al. (2022) propose two complex reasoning datasets that do not have rationale annotations. In these cases, we need unsupervised rationale selection methods to still build explainable QA systems.

Other works have explored multi-hop QA with only answer supervision but not rationale supervision. As in our work, Retrieve and Generate (RAG) (Lewis et al., 2020b), treats the rationale as a latent variable; however, in RAG the open retrieval-stage is to find a single document, ignoring the connections between different documents. RAG also does not produce sentence-level rationales. Other works consider sentence rationales: Glockner et al. (2020) compute a score for every sentence pair and pick the sentence pair that has the highest score as the rationale for answer predic-

[1]While we only consider the distractor QA setting in this work, HUG can be combined with a rule-based retrieval system such as BM25 to be adapted to the open-domain setting.

tion, and Atanasova et al. (2022) perform binary classification on whether individual sentences are included in the rationale with added constraints such as consistency and faithfulness. The shared limitation of these methods is that they do not capture the dependency between more than two pieces of information. HUG overcomes this limitation by performing multi-hop reasoning as document set prediction and sentence set prediction.

Outside of unsupervised methods, Chen et al. (2019) propose a semi-supervised method which collects silver rationale annotations. However, their method is limited to bridge-based questions, which is only one form of multi-hop reasoning; the other types can be found in Trivedi et al. (2022).

Rationales as latent variables. A focus for rationale methods in NLP outside of multi-hop QA has been identifying subsets of input tokens to justify decisions. For text classification, Lei et al. (2016), Bastings et al. (2019), and Chen and Ji (2020) frame rationales as minimal subsets of input tokens. For multi-hop QA, where input tokens are too granular a representation for rationales, treating sentences as rationales within long documents leads to the challenges of hierarchical selection and the representation of long documents; both of which we address with HUG.

Outside of using input tokens for rationales, Zhou et al. (2020) assume rationales take the form of unconstrained text. While flexible, this approach leads to computationally expensive training methods. Therefore, we constrain rationales to be a set of sentences from given documents, which both accommodates the production of useful intermediate reasoning steps and keeps training tractable.

Unsupervised retrieval. A task closely related to our setting (i.e., no access to rationale supervision) is unsupervised retrieval, which searches for sentences relevant to the questions but does not predict answers. For example, one could apply Yadav et al. (2019), Yadav et al. (2020), Zhao et al. (2021) and Xu et al. (2021) to first identify the rationale for a multi-hop QA example and predict an answer based only on the rationale rather than on the entire document, so that the answer prediction is more constrained. However, HUG may be preferred over these unsupervised retrieval methods because they assume specific types of QA formats (Xu et al., 2021), but HUG works on any type of QA problems, and 2) while other works also propose to model rationales as a latent variable, we additionally introduce the hierarchical structure in our probabilistic model, enabling efficient inference.

x: Emily Beecham is best known for her role in a television series whose second season premiered on what date?

d_1: **[1]** *Emily Beecham is an English-American actress.* **[2]** *She is best known for her role in the AMC television series "**Into the Badlands**"* **[3]** In 2011, she received the Best Actress award at the London Independent Film Festival.

d_2: **[4]** ***Into the Badlands*** *is an American television series that premiered on AMC November 15, 2015* **[5]** The series features a story about a warrior and a young boy who journey through a dangerous feudal land together seeking enlightenment. **[6]** *AMC renewed the show for a 10-episode second season, which premiered on March 19, 2017.* **[7]** On April 25, 2017, AMC renewed the series for a 16-episode third season.'

$\boldsymbol{z}$: [1], [2], [4], [6]

y: March 19, 2017

Figure 2: A QA example. The rationale $\boldsymbol{z}$ used to derive the answer is highlighted in *blue italics*, the document-level interaction is highlighted in ***red boldface***, and the sentence-level interaction (i.e., coreference resolution) is highlighted in underline. HUG models dependencies both between documents and between sentences within a document, thus being equipped with the capacity to perform multi-hop reasoning.

3 Generative Multi-Hop QA

In the standard multi-hop QA setting, an example consists of a question x, a set of documents D, and an answer y. Within D, some documents are relevant to the question, while the others are distractors. Explainable multi-hop QA models predict a rationale $\boldsymbol{z}$, a minimal set of sentences across the relevant documents, in addition to predicting the answer y. We show a multi-hop QA example (with distracting documents omitted) in Figure 2.

3.1 Model

We propose the following generative model for multi-hop QA. Given the question x, we first select a subset of documents $\boldsymbol{d} = \{d_1, d_2, \ldots\} \subseteq D$. Next, within each document d_i, we select a subset of sentences $\boldsymbol{z}_i$. Finally, conditioned on the union of sentence sets from each document, $\boldsymbol{z} = \cup_i \boldsymbol{z}_i$, we generate an answer y. The only assumption we make in the model is that sentence sets are selected independently among documents. Formally, we

write the model as,

$$p(\boldsymbol{d}, \boldsymbol{z}, y \mid x) = p(\boldsymbol{d} \mid x) \quad (1)$$
$$\cdot \prod_i p(\boldsymbol{z}_i \mid d_i, x) \quad (2)$$
$$\cdot p(y \mid \boldsymbol{z}, x). \quad (3)$$

We refer to Eq. 1 as the document set selection model, Eq. 2 as the sentence set selection model, Eq. 3 as the answer generation model.

Document Set Selection We select a set of documents $\boldsymbol{d}$ by directly parameterizing a distribution over all valid document sets. We rely on a document set scoring function $f(\boldsymbol{d}, x)$, which captures both the relevance of the document set $\boldsymbol{d}$ to the question x, as well as the dependencies among the documents in the set. The document set selection model is given by

$$p(\boldsymbol{d} \mid x) \propto \exp(f(\boldsymbol{d}, x)).$$

This distribution is globally normalized over all valid subsets of documents D, requiring the evaluation of the document scoring function f on all valid document subsets. Document set validity is dataset specific, and is discussed in Section 4.

For efficiency, the document set scoring function f first computes embeddings of each document in the set $\boldsymbol{d}$ independently, then combines them with a neural network (MLP). Formally, let emb : $\mathcal{V}^* \rightarrow \mathbb{R}^n$ be an embedding function that maps a sequence of text to an n-dimension vector, where $\mathcal{V}$ is the vocabulary. The document set scoring function is given by

$$f(\boldsymbol{d}, x) = \text{MLP}(\text{emb}(d_1, x), \text{emb}(d_2, x), \ldots).$$

We provide the details of the MLP in Appendix A and the details of the embedding function below, as part of the sentence selection model description.

Sentence Set Selection Within each document d_i, we select $\boldsymbol{z}_i \in \mathcal{P}(d_i)$, a power set of all sentences in d_i. We rely on a sentence set scoring function $g(\boldsymbol{z}_i, x)$, similar to the document set scoring function, which captures all relationships between selected sentences and the question. The sentence set selection model is given by

$$p(\boldsymbol{z}_i \mid d_i, x) \propto \exp(g(\boldsymbol{z}_i, x)),$$

which is globally normalized over all valid subsets of sentences in the document d_i.

Computing $p(\boldsymbol{z}_i \mid d_i, x)$ requires enumerating all sentences subsets, which is intractable. We instead extend the approach of Li et al. (2022), which obtains document and contextual sentence representations in a single encoding step. We insert a special `[SPC]` token at the beginning of each sentence, shown here at positions $k_1, k_2, \ldots,$

$$u = \underset{0}{\texttt{[CLS]\{x\}[SEP]}} \quad \underset{k_1}{\texttt{[SPC]}\ldots} \quad \underset{k_2}{\texttt{[SPC]}\ldots.}$$

We then obtain sentence subset $\text{emb}(\boldsymbol{z}_i)$ embeddings by feeding this to an encoder-only model such as BERT (Devlin et al., 2019) and taking the average of the contextual embeddings of the special tokens corresponding to the sentences in $\boldsymbol{z}_i$:

$$\text{emb}(\boldsymbol{z}_i, x) = \frac{1}{|\boldsymbol{z}_i|} \sum_{j \in \boldsymbol{z}_i} \text{encoder}(u)_{k_j}.$$

Finally, let v be a learnable vector, $g(\boldsymbol{z}_i, x) = v^T \text{emb}(\boldsymbol{z}_i, x)$ In practice, we note that encoder methods have a maximum input length, which can prevent full document encodings. We provide the details of long document encoding in Appendix B. We also only consider subsets of up to a fixed max length.

Answer Generation Parameterization of the answer generation model, $p(y \mid \boldsymbol{z}, x)$, is done using a sequence-to-sequence model where the question and rational are fed to an encoder, and that answer is generated. This process is complicated by the fact that answers can take on different forms, depending on specific QA tasks such as Boolean QA, multiple-choice QA, extractive QA, and abstractive QA, etc. We can therefore use a sequence-to-sequence model such as BART (Lewis et al., 2020a) and T5 (Raffel et al., 2020), and provide prompt templates for the different variants of the task, given in the next section.

3.2 Training and Inference

To learn an explainable multi-hop QA system, HUG optimizes an approximation of the marginal likelihood. The marginal likelihood,

$$\mathcal{L}(\theta) = \log \sum_{\boldsymbol{d}, \boldsymbol{z}} p_\theta(y, \boldsymbol{z}, \boldsymbol{d} \mid x),$$

is intractable, as it requires computing $p(y \mid \boldsymbol{z}, x)$ under the answer generation model for every valid

set of sentences across documents. We instead optimize a top-K Viterbi approximation of the marginal likelihood. Given

$$\mathcal{S}^k = \arg \operatorname*{topk}_{\boldsymbol{d}} p_\theta(\boldsymbol{d} \mid x)$$
$$\mathcal{S}^k_{\boldsymbol{d}} = \arg \operatorname*{topk}_{\boldsymbol{z}} \prod_i p_\theta(\boldsymbol{z}_i \mid d_i, x),$$

we use the following approximation of the marginal likelihood as our training objective:

$$\mathcal{L}(\theta) \approx \log \sum_{\boldsymbol{d} \in \mathcal{S}^k} \sum_{\boldsymbol{z} \in \mathcal{S}^k_d} p_\theta(y, \boldsymbol{z}, \boldsymbol{d} \mid x).$$

At test time, we must choose the best documents and rationales. Similar to training, we first choose the most likely pair of documents from S^1, then the most likely rationale from $S^1_{\boldsymbol{d}}$. Finally, for span-based QA, we generate an answer by performing greedy search on the answer generation model $p(y|\boldsymbol{z}, x)$; for Boolean QA or multiple-choice QA, we normalize the answers between different choices and take $\arg\max_y p(y|\boldsymbol{z}, x)$.

4 Experimental Setup

Datasets and Their Representations. We evaluate HUG on four multi-hop QA datasets: HotpotQA in the distractor setting (Yang et al., 2018), MuSiQue with answerable questions (Trivedi et al., 2022), FEVER (Thorne et al., 2018), and MultiRC (Khashabi et al., 2018). HotpotQA and MuSiQue are extractive QA datasets that require reasoning over multiple Wikipedia documents and identifying a span of text as the answer. In HotpotQA, each example contains ten candidate documents, and we must identify exactly two documents ($|D| = 2$) that are relevant. FEVER is a fact checking dataset that requires verifying claims made based on Wikipedia articles ($|D| = 1$). MultiRC is a multiple-choice QA dataset collected from diverse sources of documents including narrative stories and news articles, and their questions can only be answered by reasoning over multiple sentences ($|D| = 1$). Unlike conventional multi-choice tasks (Lai et al., 2017; Richardson et al., 2013), MultiRC does not pre-specify the number of correct answer choices, resulting in a more challenging setting. For FEVER and MultiRC, we consider the ERASER version (DeYoung et al., 2020), where rationale annotations are made cleaner and evaluation metrics are provided.

BQA	In	*A claim to be verified is that* {x} *We have following facts:* {$\boldsymbol{z}$}
	Out	*The claim is thus* {supported/refuted}.
	In	*A claim to be verified is that* Steve Wozniak designed homes. *We have following facts:* Steve Wozniak primarily designed the 1977 Apple II , known as one of the first highly successful mass-produced microcomputers.
	Out	*The claim is thus* refuted.
MCQ	In	*Question:* {x} `[SEP]` {$\boldsymbol{z}$}
	Out	*Answer:* {y_1} ({correct/wrong}) `[SEP]` *Answer:* {y_2} ({correct/wrong}) `[SEP]` ...
	In	*Question:* Name few objects said to be in or on Allan 's desk. `[SEP]` Opening a side drawer, he took out a piece of paper and his inkpot.
	Out	*Answer:* Eraser (wrong) `[SEP]` *Answer:* Inkpot (correct) `[SEP]` *Answer:* Pen (correct)
EQA	In	{x} `[SEP]` {$\boldsymbol{z}$}
	Out	{y}
	In	Which American railroad, located in Southwestern Montana and Idaho, was backed by the Northern Pacific Railway? `[SEP]` The Gilmore and Pittsburgh Railroad (G&P), now defunct, was an American railroad located in southwestern Montana and east-central Idaho.
	Out	Gilmore and Pittsburgh Railroad

Table 1: Seq-to-Seq Prompt templates for QA examples for FEVER as Boolean QA (BQA), MultiRC as multiple-choice QA (MCQ), and HotpotQA as extractive QA (EQA). Templates are in purple cells, followed by specific examples in green cells. Template keywords are highlighted in *red italics*.

In Table 1, we demonstrate how to convert QA examples of these three datasets to a natural language prompt format (Brown et al., 2020). In FEVER, a claim x needs to be classified as whether it is supported (1) or refuted (0) given the accompanying documents D. For MultiRC, there is a varying number of choices per example, and an unknown number of the choices is correct; we stack all answer choices attached with their truth values as outputs to supervise the model. Finally, extractive QA can naturally be formulated as a text-to-text problem, where the input is x `[SEP]` $\boldsymbol{z}$, and the output is y.

Metrics and Comparison Systems. We compute F1 scores for rationale and document selection, and answer prediction. F1 scores for rationales are computed at the sentence level. Because the three QA datasets are in different formats, F1 scores for answers are computed differently. For extractive QA, F1 scores are computed at the token level for the answer spans. For Boolean QA and multiple-choice QA, F1 scores measure categorical answers.

For each dataset, we compare to (1) state-of-the-

art approaches that require no rationale supervision and (2) at least one fully supervised method (i.e., answers and rationales available for training). The latter provides an upper bound on performance.

– **On HotpotQA and MuSiQue**, we compare to a rule-based approaches, BM25, and RAG (Lewis et al., 2020b) as unsupervised baselines. We note that RAG only performs document-level retrieval, and therefore its current form cannot be directly applied to identifying sentence-level rationales. We modify RAG to treat a sentence as a document, and at inference we take the top-3 sentences to be the rationale as it results in the highest sentence F1 scores. For fair comparison, we parameterize RAG in the same way as we parameterize HUG. We also consider a semi-supervised approach – CHAIN (Chen et al., 2019); they assume no access to gold rationale annotations and supervise their model on silver rationales produced with external entity taggers. The fully supervised system we consider is SAE (Tu et al., 2020). Both CHAIN and SAE use RoBERTa-large as sentence encoders.

– **On FEVER and MultiRC**, we also compare to RAG as an unsupervised baseline (predicting top-2 sentences for MultiRC, and top-1 sentence for FEVER). Additionally, we consider diagnostics-guided explanation generation (DIAGNOSTICS) (Atanasova et al., 2022) and faithful rationales (FAITHFUL) (Glockner et al., 2020). Both of these methods have two variants – one trained with rationale supervision (denoted by $\mathbb{RS}$-*) and the other trained without rationale annotations (denoted by $\mathbb{RU}$-*). On MultiRC, We also compare to WT5 (Narang et al., 2020).

Implementation and Hyperparameters. We test HUG with language models of both small and large sizes. For the small version (HUG-Small), we use distilBERT (Sanh et al., 2019) as the encoder and BART-base as the seq2seq model. For the large version (HUG), we use RoBERTa-large as the encoder and BART-large as the seq2seq model. We only test RAG in the small version.

We implement HUG with Hugging Face Transformers (Wolf et al., 2020). We perform grid search with learning rates {5e-6, 1e-5, 2e-5} and batch sizes {2,4,8,16} for both HUG and RAG. We train our system for 3 epochs for HotpotQA and 5 epochs for the other three datasets. We warm up the learning rate with first 10% examples. We choose the checkpoint that has the highest answer F1 score on the validation set. We consider rationales up to four sentences for HotpotQA and rationales up to three sentences for the other datasets. Finally, we take S^{10} and S^{9}_{d} for HotpotQA and S^{80}_{d} for MultiRC. Because we remove rationales whose sentences are not contiguous in FEVER, we are able to compute the exact likelihood without top-k sampling.

5 Results

HotpotQA. We summarize the results on HotpotQA in Table 2. HUG-Small outperforms the best unsupervised approach RAG-Small by 18 sentence F1 points, demonstrating superior multi-hop reasoning abilities. HUG-Small, despite having fewer parameters, outperforms the semi-supervised CHAIN on predicting rationales and is comparable to CHAIN on predicting answers. While CHAIN explicitly exploits the heuristics used in the data collection process for HotpotQA, HUG-Small is able to learn such heuristics in a fully automatic way. Finally, the gap between HUG and SAE, a fully supervised method that is given both rationale and answer annotations, remains large.

MuSiQue. Table 2 shows that HUG-Small is better at both identifying rationales and predicting answers than RAG-Small, the best-performing unsupervised baseline. Due to the difficulty of MuSiQue, it is harder for an unsupervised method to learn to select rationales – the gap between the supervised method and the best unsupervised method on this dataset is greater than the gap on HotpotQA.

FEVER We summarize the results on FEVER in Table 3. In terms of selecting rationales in an unsupervised manner, HUG-Small outperforms $\mathbb{RU}$-DIAGNOSTICS, performs similarly to RAG-Small, and underperforms $\mathbb{RU}$-FAITHFUL by a small margin. Because FEVER mostly only requires single-hop reasoning, HUG-Small does not improve over the previous methods. On predicting answers, HUG-Small outperforms $\mathbb{RU}$-FAITHFUL but underperforms RAG-Small. Compared to the supervised versions of $\mathbb{RS}$-DIAGNOSTICS and $\mathbb{RS}$-FAITHFUL, which have access to both the answers and rationales during training, the gap between HUG-Small's and their rationale scores (sentence F1) understandably remains large.

MultiRC Table 3 shows that HUG-Small outperforms all comparison models including RAG-Small – the best competing approach without ratio-

	#Params	HotpotQA **Sent F1**	HotpotQA Ans F1	MuSiQue **Sent F1**	MuSiQue Ans F1
BM25	-	40.5	-	12.9	-
RAG-Small	221M	49.0	62.8	32.0	24.2
HUG-Small	221M	67.1	66.8	34.2	25.1
HUG	761M	**72.5**	73.5	**44.4**	39.1
CHAIN (semi-supervised)	355M	64.5	66.0	-	-
SAE (supervised)	790M	87.4	80.8	75.2	52.3

Table 2: Performance comparison on predicting rationales and answers on HotpotQA and MuSiQue.

	#Params	FEVER **Sent F1**	FEVER Ans F1	MultiRC **Sent F1**	MultiRC Ans F1
$\mathbb{RU}$-FAITHFUL	110M	83.8	90.6	27.5	67.7
$\mathbb{RU}$-DIAGNOSTICS	110M	56.1	-	38.1	-
RAG-Small	221M	80.7	92.0	44.9	70.0
HUG-Small	221M	81.5	91.6	48.4	74.3
HUG	761M	**84.1**	94.3	**55.6**	75.5
$\mathbb{RS}$-FAITHFUL (supervised)	110M	91.4	91.1	46.1	67.4
$\mathbb{RS}$-DIAGNOSTICS (supervised)	110M	94.4	89.7	79.4	71.7

Table 3: Performance comparison on predicting rationales and answers on Eraser-FEVER and Eraser-MultiRC. HUG has more parameters due to its use of a seq2seq model for answer generation.

		Sent F1	Doc F1	Ans F1
Comparison	HUG-Ind	**78.9**	**92.9**	64.8
	HUG	78.1	91.1	**69.7**
Bridge	HUG-Ind	55.2	68.6	71.6
	HUG	**71.0**	**87.3**	**75.7**
Combined	HUG-Ind	60.0	73.4	69.1
	HUG	**72.5**	**88.0**	**73.5**

Table 4: Rationale selection performance broken down by different types of reasoning.

nale supervision – by 3.5 sentence F1 points and 4.3 answer F1 points.

Scaling HUG to Larger Models. On all three datasets, by increasing the number of model parameters, HUG can consistently achieve better performance. Additionally, as the number of reasoning hops increases, HUG can more benefit from the larger language models – compared to HUG-Small, HUG has the least improvement on FEVER and had the most improvement on MuSiQue.

6 Analysis

Document Dependencies HUG explicitly models the dependencies between documents for multi-hop reasoning. We consider independent document selection [2] to see whether this dependencies is necessary on the HotpotQA dataset.

To understand how document modeling impact rationale selection performance, we break the performance down by the reasoning types proposed in Yang et al. (2018): comparison-based reasoning and bridge-based reasoning. In comparison-based reasoning, relevant documents independently contribute to the answer, whereas for bridge-based reasoning, relevant documents require connections to previously selected documents. Table 4 summarizes answer F1 scores, document F1 scores, and sentence F1 scores. While HUG-Ind is slightly better at comparison-based reasoning than the joint model, it fails at bridge-based reasoning; this result thus confirms the necessity of modeling the dependency between documents. We also note that HUG-Ind and HUG have similar performance in predicting answers, but the gap between how accurate they select rationales is large, suggesting that HUG-Ind often derives answers with wrong reasoning. Overall, HUG is better than HUG-Ind

[2] We provide the detail of training and testing independent document selection model in Appendex D.

Q: When **Copsi** was made **earl of Northumbria** he went to reside in a town at the confluence of which **two rivers**?

Document A, Copsi:
Copsi survived Tostig's defeat at Stamford Bridge, and when William the Conqueror prevailed at Hastings he travelled, in March 1067, to pay William homage at Barking (where William was staying while his tower was being constructed in London). *In return, William made **Copsi earl of Northumbria** and sent him back to York.*

Document B, York:
York is a historic walled city at the confluence of the rivers Ouse and Foss in North Yorkshire, England. The municipality is the traditional county town of the historic county of Yorkshire to which it gives its name.

Document C, Two Rivers Press:
Two Rivers Press is an independent publishing house, based in the English town of Reading. Two Rivers Press was founded in 1994 by Peter Hay (1951–2003).

A: Ouse and Foss

Figure 3: A HotpotQA example where there is a dependency between two supporting documents, and thus selecting the second document independent of the first one results in insufficient information. Correct rationale is highlighted in *blue italics*. Entity overlaps between questions and documents are in **red boldface**. HUG-Ind's predicted Documents B and C, whose reasoning remains at the surface level as they share the most entities with the question. HUG predicted Documents A and B, which demonstrates its ability of understanding dependency between documents.

Answer Model	Pred Type	FEVER	MultiRC
		Sent F1	
BART	Generate	81.5	48.4
RoBERTa	Classify	82.0	20.0

Table 5: Comparison on sentence F1 scores between different parameterization choices of $P(y \mid \boldsymbol{z}, x)$.

at both predicting answers and selecting rationales.

In addition to the quantitative analysis, we also qualitatively compare the two models in Figure 3. When considering paragraphs independently, documents A and C share the most entities with the question (i.e., Copsi, earl of Northumbria, and Two Rivers), so they are more likely to lead to the answer. However, the correct documents are A and B. Deriving B not only depends on the question but also further requires knowing the information from A. Therefore, while having the independent document selection model can improve efficiency because it only performs one-step reasoning, the joint document selection model is necessary when reasoning steps depend on one another.

Role of Answer Generation HUG uses a generative model (BART) to parameterize $p(y \mid \boldsymbol{z}, x)$.

Q: What did the judge tell Mr. Thorndike about the law?
A1: *Cannot be swayed by wealth or political influences.*
A2: *The law is not vindictive.*
A3: *It was not vindictive.*
A4: It was unjust.
A5: It was vindictive.
A6: *The judge told Mr. Thorndike that the law is not vindictive. He said the law only wishes to be just. Judge said the law cannot be swayed by wealth or political influences.*

Figure 4: A test example from MultiRC that can be answered with commonsense reasoning and thus requires no accompanying documents. Correct answers are highlighted in *blue italics*.

	HUG	FAITHFUL	Ratio
Training	11,544.2	444.0	26
Inference	39.0	3.9	10

Table 6: Runtime comparison (in seconds). HUG uses 80 rationale samples at training time, and the argmax rationale at inference.

An alternative approach would be to use a classification model such as RoBERTa (Liu et al., 2019) to predict answers for FEVER and MultiRC (HotpotQA requires a generative model).

Interestingly, Table 5 shows the choice of answer model significantly impacts the ability of HUG to learn a rationale model. On FEVER, where claims cannot be verified without the corresponding rationales, BART and RoBERTa perform similarly. However, on MultiRC, where questions can often be answered without information in accompanying documents, the best Generative model outperforms the best Classification model by over 32 sentence F1 points.

Figure 4 shows an example of such a question where the answers can be guessed by the classification model using commonsense knowledge to reason about law. Generative models need to assign a high probability to every token in the answer, and we hypothesize that they make better use of the answer supervision.

Speed evaluation. While HUG obtains strong sentence F1 scores, training is more expensive because the model must consider a set of rationales for every example. In particular, the answer model $p(y \mid \boldsymbol{z}, x)$ must be run for *every* sampled $\boldsymbol{z}$ for each training example. At inference, the answer model requires only a single evaluation of $p(y \mid \boldsymbol{z}, x)$ for $\arg\max_{\boldsymbol{z}} p(\boldsymbol{z} \mid x)$. We empirically measure the runtime overhead of HUG compared to FAITHFUL on MultiRC, using 80 samples of $\boldsymbol{z}$ at

training time. We report the total training time and inference time in Table 6. Compared to FAITHFUL, HUG takes longer to train and to predict.

7 Conclusion

We present HUG, a probabilistic, principled approach for explainable multi-hop reasoning without rationale supervision. HUG explicitly models multi-hop reasoning by considering the dependency between documents and between sentences within a document. Experimental results demonstrate that HUG outperforms other state-of-the-art methods that do not rely on rationale labels.

Ethics Statement

The goal of explainable methods is to improve the trustworthiness of systems. HUG presents a method for fine-tuning language models for selecting rationales, without rationale annotations, that exploits the knowledge already present in pretrained language models. While this has the potential of improving the trustworthiness of the model, it may also reinforce existing harmful biases in the language model.

Acknowledgement

AR and JC are supported by a Sloan Fellowship, NSF CAREER #2037519, NSF #2242302, and NSF #1901030. CC and WZ are supported by NSF #1815455.

References

Pepa Atanasova, Jakob Grue Simonsen, Christina Lioma, and Isabelle Augenstein. 2022. Diagnostics-guided explanation generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 10445–10453.

Jasmijn Bastings, Wilker Aziz, and Ivan Titov. 2019. Interpretable neural predictions with differentiable binary variables. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2963–2977, Florence, Italy. Association for Computational Linguistics.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Hanjie Chen and Yangfeng Ji. 2020. Learning variational word masks to improve the interpretability of neural text classifiers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4236–4251, Online. Association for Computational Linguistics.

Jifan Chen and Greg Durrett. 2019. Understanding dataset design choices for multi-hop reasoning. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4026–4032, Minneapolis, Minnesota. Association for Computational Linguistics.

Jifan Chen, Shih-ting Lin, and Greg Durrett. 2019. Multi-hop question answering via reasoning chains. *arXiv preprint arXiv:1910.02610*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Jay DeYoung, Sarthak Jain, Nazneen Fatema Rajani, Eric Lehman, Caiming Xiong, Richard Socher, and Byron C. Wallace. 2020. ERASER: A benchmark to evaluate rationalized NLP models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4443–4458, Online. Association for Computational Linguistics.

Yuwei Fang, Siqi Sun, Zhe Gan, Rohit Pillai, Shuohang Wang, and Jingjing Liu. 2020. Hierarchical graph network for multi-hop question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8823–8838, Online. Association for Computational Linguistics.

Mor Geva, Daniel Khashabi, Elad Segal, Tushar Khot, Dan Roth, and Jonathan Berant. 2021. Did aristotle use a laptop? a question answering benchmark with implicit reasoning strategies. *Transactions of the Association for Computational Linguistics*, 9:346–361.

Max Glockner, Ivan Habernal, and Iryna Gurevych. 2020. Why do you think that? exploring faithful sentence-level rationales without supervision. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1080–1095, Online. Association for Computational Linguistics.

Dirk Groeneveld, Tushar Khot, Mausam, and Ashish Sabharwal. 2020. A simple yet strong pipeline for HotpotQA. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8839–8845, Online. Association for Computational Linguistics.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S. Weld, Luke Zettlemoyer, and Omer Levy. 2020. SpanBERT: Improving pre-training by representing and predicting spans. *Transactions of the Association for Computational Linguistics*, 8:64–77.

Daniel Khashabi, Snigdha Chaturvedi, Michael Roth, Shyam Upadhyay, and Dan Roth. 2018. Looking beyond the surface: A challenge set for reading comprehension over multiple sentences. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 252–262, New Orleans, Louisiana. Association for Computational Linguistics.

Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. 2017. RACE: Large-scale ReAding comprehension dataset from examinations. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 785–794, Copenhagen, Denmark. Association for Computational Linguistics.

Tao Lei, Regina Barzilay, and Tommi Jaakkola. 2016. Rationalizing neural predictions. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 107–117, Austin, Texas. Association for Computational Linguistics.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020a. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020b. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Advances in Neural Information Processing Systems*, volume 33, pages 9459–9474. Curran Associates, Inc.

Xin-Yi Li, Wei-Jun Lei, and Yu-Bin Yang. 2022. From easy to hard: Two-stage selector and reader for multi-hop question answering. *arXiv preprint arXiv:2205.11729*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Sewon Min, Eric Wallace, Sameer Singh, Matt Gardner, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2019. Compositional questions do not necessitate multi-hop reasoning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4249–4257, Florence, Italy. Association for Computational Linguistics.

Sharan Narang, Colin Raffel, Katherine Lee, Adam Roberts, Noah Fiedel, and Karishma Malkan. 2020. Wt5?! training text-to-text models to explain their predictions. *CoRR*, abs/2004.14546.

Peng Qi, Xiaowen Lin, Leo Mehr, Zijian Wang, and Christopher D. Manning. 2019. Answering complex open-domain questions through iterative query generation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2590–2602, Hong Kong, China. Association for Computational Linguistics.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Matthew Richardson, Christopher J.C. Burges, and Erin Renshaw. 2013. MCTest: A challenge dataset for the open-domain machine comprehension of text. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 193–203, Seattle, Washington, USA. Association for Computational Linguistics.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *ArXiv*, abs/1910.01108.

Yixuan Tang, Hwee Tou Ng, and Anthony Tung. 2021. Do multi-hop question answering systems know how to answer the single-hop sub-questions? In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 3244–3249, Online. Association for Computational Linguistics.

Mokanarangan Thayaparan, Marco Valentino, and André Freitas. 2020. A survey on explainability in machine reading comprehension. *arXiv preprint arXiv:2010.00389*.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018. FEVER: a large-scale dataset for fact extraction and VERification. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 809–819, New Orleans, Louisiana. Association for Computational Linguistics.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2022. MuSiQue: Multi-hop questions via single-hop question composition. *Transactions of the Association for Computational Linguistics*.

Ming Tu, Kevin Huang, Guangtao Wang, Jing Huang, Xiaodong He, and Bowen Zhou. 2020. Select, answer and explain: Interpretable multi-hop reading comprehension over multiple documents. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 9073–9080.

Ming Tu, Guangtao Wang, Jing Huang, Yun Tang, Xiaodong He, and Bowen Zhou. 2019. Multi-hop reading comprehension across multiple documents by reasoning over heterogeneous graphs. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2704–2713, Florence, Italy. Association for Computational Linguistics.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed H. Chi, Quoc V Le, and Denny Zhou. 2022. Chain of thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*.

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

Sarah Wiegreffe and Ana Marasovic. 2021. Teach me to explain: A review of datasets for explainable natural language processing. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 1)*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Weiwen Xu, Yang Deng, Huihui Zhang, Deng Cai, and Wai Lam. 2021. Exploiting reasoning chains for multi-hop science question answering. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 1143–1156, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Vikas Yadav, Steven Bethard, and Mihai Surdeanu. 2019. Quick and (not so) dirty: Unsupervised selection of justification sentences for multi-hop question answering. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2578–2589, Hong Kong, China. Association for Computational Linguistics.

Vikas Yadav, Steven Bethard, and Mihai Surdeanu. 2020. Unsupervised alignment-based iterative evidence retrieval for multi-hop question answering. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4514–4525, Online. Association for Computational Linguistics.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380, Brussels, Belgium. Association for Computational Linguistics.

Xinyan Velocity Yu, Sewon Min, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2022. Crepe: Open-domain question answering with false presuppositions. *arXiv preprint arXiv:2211.17257*.

Hongming Zhang, Xinran Zhao, and Yangqiu Song. 2020. WinoWhy: A deep diagnosis of essential commonsense knowledge for answering Winograd schema challenge. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5736–5745, Online. Association for Computational Linguistics.

Chen Zhao, Chenyan Xiong, Jordan Boyd-Graber, and Hal Daumé III. 2021. Distantly-supervised dense retrieval enables open-domain question answering without evidence annotation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 9612–9622, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Wangchunshu Zhou, Jinyi Hu, Hanlin Zhang, Xiaodan Liang, Maosong Sun, Chenyan Xiong, and Jian Tang. 2020. Towards interpretable natural language understanding with explanations as latent variables. *Advances in Neural Information Processing Systems*, 33:6803–6814.

A Document scoring function

Let MLP : $\mathbb{R}^{3n} \to \mathbb{R}$ be a multilayer perception:

$$\text{MLP}(x) = W_2\text{ReLU}(W_1 x).$$

The document set scoring function is given by:

$$f(\boldsymbol{d}, x) = \text{MLP}(\text{emb}(d_1, x), \ldots).$$

For document pairs, we specialize this to

$$f(\boldsymbol{d}, x) \\ = \text{MLP}([\text{emb}(d_1, x), \text{emb}(d_2, x), s(d_1, d_2)])$$

where

$$s(d_1, d_2) = |\text{emb}(d_1, x) - \text{emb}(d_2, x)|.$$

Criteria	Value	Sentence F1
Answer F1	66.81	**67.05**
Answer EM	53.51	**67.05**
Answer NLL	3.01	65.82

Table 7: Sentence F1 scores from checkpoints that are chosen based on different criteria.

For extending this parameterization to large document sets, we could use a similar parameterization to the sentence set scoring function:

$$f(\boldsymbol{d}, x) = \text{MLP}(\sum_i \text{emb}(d_i, x)).$$

B Encoding long documents

Transformer-based text encoders can only accept inputs shorter than a fixed length (e.g., 512 tokens). To address this limitation, we partition documents into slices of m sentences and compute the embedding for each slice individually. We denote a slice for a document d as $d^{i:j}$ that starts at the ith sentence and ends before the jth sentence. Let $i \in \text{range}(0, |d|, p)$, we approximate $\text{emb}(d)$ by the following aggregation,

$$\text{emb}(d) \approx \left\lfloor \frac{|d|}{m} \right\rfloor \sum_i \text{emb}(d^{i:i+p}).$$

We set the slice length m purely based on whether the longest slice is under 512 tokens. m is set to 3 for HotpotQA, 5 for FEVER, and 9 for MultiRC.

C Model selection

In the unsupervised sentence selection setting, we cannot perform model selection by choosing the model with the highest validation sentence F1 score. Instead, we must rely on answer evaluation measures: validation answer F1, answer EM, or likelihood. We train HUG for three epochs, checkpoint every 2500 steps, and evaluate sentence F1 for the checkpoint with the best validation performance measure. The results of these selection methods are presented in Table 7. We find that performing model selection via both answer EM and answer F1 results in the best sentence F1, but the differences between different metrics are minor.

D Independent document selection model

For independent document selection, we train a different document selection model that factors as

$$p(\boldsymbol{d} \mid x) = \prod_{d \in \boldsymbol{d}} p(d \mid x).$$

Q: Watertown International Airport and Blue Grass Airport, are in which country?

Document A, Blue Grass Airport:
Blue Grass Airport is a public airport in Fayette County, Kentucky, 4 miles west of downtown Lexington.
Document B, Watertown International Airport:
Watertown International Airport is a county owned, public use airport located in Jefferson County, New York, United States.

A: United States

Q: Who is also an actor, Luis Llosa or Ron Howard?

Document A, Luis Llosa:
Luis Llosa (born 1951) is a Peruvian film director.
Document B, Ronald William Howard:
Ronald William Howard (born March 1, 1954) is an American actor and filmmaker.

A: Ronald William Howard

Figure 5: Dataset Shortcuts. Two HotpotQA examples that do not need both documents to derive the answers.

Exact marginalization of $\boldsymbol{d}, \boldsymbol{z}$ is still intractable. We thus only marginalize over $\mathcal{S}^5$ and $\mathcal{S}^5_{\boldsymbol{d}}$. At inference, we choose

$$\boldsymbol{d} = \arg \operatorname*{topk}_d p(d \mid x),$$

where k is the number of documents pre-specified by the task.

E Revealing dataset shortcuts with HUG

We show that HUG is able to discover examples in which answers can be derived with reasoning shortcuts. Yang et al. (2018) claim that all HotpotQA examples require reasoning over two documents, but we identify a number of examples that fail this property with the following steps. First, we look for the examples that HUG correctly predicts the answers but incorrectly predicts the rationales. Of those examples, we look for a subset where only one document is correctly selected. Because if conditioning on the one document that can already lead to the answer, the other document is redundant. Finally, we manually go through the filtered examples and find that many of the questions can be answered with one documents. Figure 5 shows two types of reasoning shortcuts found by us. The first question implies both airports are in the same country, and thus looking up one of the airports is sufficient. In the second question, Document B alone contain the correct answer.