MuSiQue: Multi-hop Questions via Single-hop Question Composition

Harsh Trivedi† Niranjan Balasubramanian† Tushar Khot‡ Ashish Sabharwal‡

† Stony Brook University, Stony Brook, U.S.A. 
{hjtrivedi,niranjan}@cs.stonybrook.edu
‡ Allen Institute for AI, Seattle, U.S.A. 
{tushark,ashishs}@allenai.org

Abstract

Can we create a question answering (QA) dataset that, by construction, requires proper multi-hop reasoning? This goal has been surprisingly elusive. We introduce a bottom-up approach that systematically selects composable pairs of single-hop questions that are connected, i.e., where one reasoning step requires information from the other. This bottom-up approach allows greater control over the properties of the resulting k-hop questions. We add stringent filters and other mechanisms targeting connected reasoning, including minimizing many forms of train-test leakage, improved distractor contexts, and contrasting unanswerable questions at the sub-question level. We use this process to construct MuSiQue-Ans, a new multihop QA dataset with 25K 2-4 hop questions, built using seed questions from 5 existing single-hop datasets. Our experiments demonstrate that MuSiQue-Ans is challenging for state-of-the-art QA models significantly harder than existing datasets (3x human-machine gap in a comparable setting), and substantially less cheatable (e.g., a single-hop model is worse by 30 F1 pts). We also build a more challenging dataset, MuSiQue-Full, consisting of answerable and unanswerable contrast question pairs, where model performance drops further by 14 F1 pts.

1 Introduction

Multi-hop QA datasets are designed to support the development and evaluation of models that perform multiple steps of reasoning in order to answer a question. Recent work, however, shows that on existing datasets, models often need not even connect information across all supporting facts,2 because they can exploit reasoning shortcuts and other artifacts to find the correct answers and obtain high scores (Min et al., 2019a; Chen and Durrett, 2019; Trivedi et al., 2020).

Fig. 1 illustrates a key challenge in ensuring connected reasoning. Both questions in the figure can be seen as requiring multiple linked steps of reasoning. They both appear to require a model to identify Indonesia as the country where Triskati university is located. However, this requirement, by itself, is not sufficient to ensure connected multi-hop reasoning. E.g., Q′ (right) can be answered without finding the answer to Q1′ (there is only one country that got independence in Dec 1949).

Fig. 1: Generating connected multi-hop questions by composing carefully chosen pairs of single-hop questions. Left: A connected question that forces models to reason through all intended hops. Right: A potential question filtered out by our approach for not requiring connected reasoning; it can be answered using just Q2′ without knowing the answer to Q1′ (there is only one country that got independence in Dec 1949).

1 Code and datasets are available at https://github.com/stonybrooknlp/musique.

2 e.g. don’t even use information from one supporting fact to select another.
where a model arrives at the correct answer without using all supporting facts, as Disconnected Reasoning (Trivedi et al., 2020). While this enables filtering or automatically transforming existing datasets (Trivedi et al., 2020), in this work we ask: How can we construct a new multi-hop dataset that, by design, enforces connected reasoning? We make two main contributions:

1) A new dataset construction approach: We introduce a bottom-up process for building challenging multi-hop reading comprehension QA datasets by carefully selecting and composing single-hop questions obtained from existing datasets. The key ideas behind our approach are: (i) Composing multihop questions from a large collection of single-hop questions, which allows a systematic exploration of a vast space of candidate multi-hop questions. (ii) Applying a stringent set of filters that ensure no sub-question can be answered without finding the answer to the previous sub-questions it is connected to. (iii) Reducing train-test leakage at the level of each single-hop question, thereby mitigating the impact of simple memorization tricks. (iv) Adding distractor contexts that cannot be easily identified. (v) Creating unanswerable multi-hop questions by making a constituent single-hop question unanswerable.

2) A new challenge dataset and empirical analysis: We build a new multi-hop QA dataset, MuSiQue-Ans (abbreviated as \( \mathcal{Q}_A \)-Ans), with \( \sim 25K \) 2-4 hop questions with six different composition structures (cf. Table 1). We demonstrate that \( \mathcal{Q}_A \)-Ans is more challenging and less cheatable than two prior multi-hop reasoning datasets, HotpotQA (Yang et al., 2018) and 2WikiMultihopQA (Ho et al., 2020). In particular, it has 3x the human-machine gap, and a substantially lower disconnected reasoning (DiRe) score which captures the extent to which a dataset can be cheated via disconnected reasoning (Trivedi et al., 2020). We also show how various features of our dataset construction pipeline help increase dataset difficulty and reduce cheatability. Lastly, by incorporating the notion of insufficient context (Rajpurkar et al., 2018; Trivedi et al., 2020), we also release a variant of our dataset, \( \mathcal{Q}_A \)-Full, having \( \sim 50K \) multi-hop questions which form contrasting pairs (Kaushik et al., 2019; Gardner et al., 2020) of answerable and unanswerable questions. \( \mathcal{Q}_A \)-Full is even more challenging and harder to cheat on.

2 Related Work

Multihop QA. MuSiQue-Ans is closest to HotpotQA (Yang et al., 2018) and 2WikiMultihopQA (Ho et al., 2020). HotpotQA was constructed by directly crowdsourcing 2-hop questions without considering the difficulty of composition and has been shown to be largely solvable without multi-hop reasoning (Min et al., 2019a; Trivedi et al., 2020). While 2WikiMultihopQA was also constructed via composition, they use a limited set of hand-authored compositional rules making it easy for large LMs. We show that MuSiQue-Ans is harder and less cheatable than both these datasets. Other multi-hop reasoning datasets (Khashabi et al., 2018; Dua et al., 2019, inter alia) focus on different challenges such as multiple modalities (Chen et al., 2020; Talmor et al., 2021), open-domain QA (Geva et al., 2021; Khot et al., 2020), fact verification (Jiang et al., 2020), science explanations (Jansen et al., 2018), and relation extraction (Welbl et al., 2018), among others. Extending our ideas to these challenges is an interesting avenue for future work.

Unanswerable QA. Prior works have used unanswerable questions for robust reasoning in single-hop (Rajpurkar et al., 2018) and multi-hop (Ferguson et al., 2020; Trivedi et al., 2020) settings. Our idea to make unanswerable multi-hop questions by removing support paragraphs is most similar to Trivedi et al. (2020). While they rely on annotations (potentially incomplete) to identify these support paragraphs, we can use the bridge entities to remove any potential support paragraphs (containing the bridge entity) and better ensure unanswerability.

Question Decomposition and Composition. Multi-hop QA datasets have been decomposed into simpler questions (Min et al., 2019b; Talmor and Berant, 2018) and special meaning representations (Wolfson et al., 2020). Our dataset creation pipeline naturally provides question decompositions, which can can help develop interpretable models (Min et al., 2019b; Khot et al., 2021).

Some recent works have also used bottom-up approaches to create multi-hop questions (Pan et al., 2021; Yoran et al., 2021) using rule-based methods. However these questions were created with the primary goal of data augmentation to improve on downstream datasets. The questions themselves haven’t been shown to be challenging.
3 Multihop Reasoning Desiderata

Multi-hop question answering can be seen as a sequence of inter-dependent reasoning steps leading to the answer. In its most general form, these reasoning steps and the dependencies can be viewed as directed acyclic graph (DAG), $G_Q$. Each node $q_i$ in this graph represents a reasoning step or a “hop”, e.g., a single-hop question in multi-hop QA or a KB relation traversal in graph-based KBQA. An edge $(q_i, q_j) \in \text{edges}(G_Q)$ indicates that the reasoning step $q_i$ relies critically on the output of the predecessor step $q_j$. For example, in Fig. 1, the single-hop question $Q2$ depends on the answer to $Q1$, and the graph $G_Q$ is a linear chain $Q1 \rightarrow Q2$.

Given this framing, a key desirable property for multi-hop reasoning is connected reasoning: 

**Analytical Intuition.** Suppose a model $M$ can answer each $q_i$ correctly with probability $p$, and it can also answer $q_i$ without the output of all its predecessor steps with probability $r \leq p$. For simplicity, we assume these probabilities are independent across various $q_i$. $M$ can correctly answer a $k$-hop question $Q$ by identifying and performing all its $k$ reasoning steps. This will succeed with probability at most $p^k$. Alternatively, as an extreme case, it can “cheat” by identifying and performing only the last step $q_k$ (the “end question”) without considering the output of $q_{k-1}$ (or other steps) at all. This could succeed with probability as much as $r$, which does not decrease with $k$ and is thus undesirable when constructing multi-hop datasets. Our goal is to create multi-hop questions that enforce connected reasoning, i.e., where $r \ll p$ and, in particular, $r < p^k$, so that models have an incentive to perform all $k$ reasoning steps.

Not surprisingly, the connected reasoning property is often not satisfied by existing datasets (Min et al., 2019a; Chen and Durrett, 2019; Trivedi et al., 2020), and never optimized for during dataset construction. As a consequence, models are able to exploit artifacts in existing datasets that allow them to achieve high scores while bypassing some of the reasoning steps, thus negating the main purpose of building multi-hop datasets. Prior work (Trivedi et al., 2020) has attempted to measure the extent of connected reasoning in current models and datasets. However, due to the design of existing datasets, their approach is only able to measure this by ablating the pre-requisites of each reasoning step, i.e., the supporting facts. Rather that only measure, we propose a method to construct multi-hop QA datasets that directly optimize for this condition.

Consider question $Q'$ (right) in Fig. 1, which can be answered in 2 steps, $Q1'$ and $Q2'$. However, the information in $Q2'$ itself is enough to uniquely identify $A2'$ from the context, even without considering $A1'$. I.e., while there is an intended dependency between $Q1'$ and $Q2'$, $Q2'$ can be answered correctly without requiring the output of its predecessor. Our approach constructs multi-hop questions that prevent this issue, and thereby, require the desired connected reasoning by composing single-hop questions where each question necessitates the answers from previous questions.

4 Connected Reasoning via Composition

The central issue we want to address is ensuring connected reasoning. Our solution is to use a bottom-up approach where we compose multihop...
questions from a large pool of single-hop questions. As we show later, this approach allows us to explore a large space of multi-hop questions and carefully select ones that require connected reasoning. Additionally, with each multi-hop question, we will have associated constituent questions, their answers and supporting paragraphs, which can help develop more interpretable models. Here we describe the high-level process and describe the specifics in the next section.

4.1 Multi-hop via Single Hop Composition

As mentioned earlier, multi-hop questions can be viewed as a sequence of reasoning steps where answer from one reasoning step is used to identify the next reasoning step. Therefore, we can use single-hop questions containing answers from other questions to construct potential multi-hop questions. E.g., in Fig. 1, Q2 mentions A1, and other questions to construct potential multi-hop question Q, hence satisfies condition (1). Consider the masked questions Q2 and Q2’ in Fig. 1. While Q2’ can easily be answered without answer A1’, Q2 can’t be answered without A1 and Q hence satisfies condition (1).

4.2 Ensuring Connected Reasoning

Given the graph $G_Q$ associated with a question $Q$, ensuring connected reasoning requires ensuring that for each edge $(q_j, q_i) \in \text{edges}(G_Q)$, arriving at answer $a_i$ using $q_i$, necessitates the use of $a_j$. In other words, without $a_j$, there isn’t sufficient information in $q_i$ to arrive at $a_i$.

The existence of such information can be probed by training a strong QA model $M$ on sub-questions $(q_i)$ with the mention of their predecessor’s answer $(a_j)$ masked out (removed). If, on held out data, the model can identify a sub-question’s answer $(a_i)$ without its predecessor’s answer $(a_j)$, we say the edge $(q_j, q_i)$ is disconnected. Formally, we say $Q$ requires connected reasoning if:

$$\forall (q_j, q_i) \in \text{edges}(G_Q) : M(q_i^{m_j}) \neq a_i$$  (1)

where $q_i^{m_j}$ denotes the subquestion formed from $q_i$ by masking out the mention of the answer $a_j$.

Consider the masked questions Q2 and Q2’ in Fig. 1. While Q2’ can easily be answered without answer A1’, Q2 can’t be answered without A1 and Q hence satisfies condition (1).

4.3 Reading Comprehension Setting

While our proposed framework makes no assumptions about the choice of the model, and is applicable to open-domain setting, we focus on the Reading Comprehension (RC) setting, where we’ve a fixed set of paragraphs as context, $C$.

In a RC setting, apart from requiring the dependence between the reasoning steps, we also want the model to depend on the context to answer each question. While this requirement seems unnecessary, previous works have shown that RC datasets often have artifacts that allow models to predict the answer without the context (Kaushik and Lipton, 2018) and can even memorize the answers (Lewis et al., 2021) due to train-test leakage. As we will show later, previous multi-hop RC datasets can be cheated via such shortcuts. To ensure the dependence between the question and context, we modify the required condition in Eqn. (1) to:

$$\forall (q_j, q_i) \in \text{edges}(G_Q) : M(q_i^{m_j}; C) \neq a_i$$

$$\land \forall q_i \in \text{nodes}(G_Q) : M(q_i; \emptyset) \neq a_i$$  (2)

In summary, we want multi-hop reading comprehension questions that satisfy condition 2 for a strong trained model $M$. If it does, we say that the question satisfies the MuSiQue condition. Our dataset construction pipeline optimizes for this condition as described next.

5 Dataset Construction Pipeline

The high-level schematic of the pipeline is shown in Fig. 2. We begin with a large set of RC single-hop questions from 5 English Wikipedia-based datasets, SQuAD (Rajpurkar et al., 2016), Natural Questions (Kwiatkowski et al., 2019), MLQA (en-en) (Lewis et al., 2020b), T-REx (ElSahar et al., 2018), Zero Shot RE (Levy et al., 2017), where instances are of the form $(q_i, p_i, a_i)$ referring to the question, associated paragraph, and the answer respectively. Then, we take the following steps:
S1. Find Good Single-Hop Questions  Even a (tolerably) small percentage of issues in single-hop questions can compound and become an (intolerably) large percentage in the composed multihop questions. To mitigate this, we remove questions that are spuriously hard — ones with extremely long or short contexts (<20 or >300 words), with long answers (>15 words), or ones whose answers are not in their contexts. There are also a small fraction with annotation errors which make them ill-defined questions but manually identifying these at scale is difficult. To avoid these, we use a conservative approach of only keeping those questions whose associated answers are found (>0 answer F1) by at least one out of a suite of five large trained QA models.

S2. Find Composable Single-Hop Pairs  To create 2-hop questions, we first collect distinct single-hop question pairs with a bridge entity. Specifically, we find pairs \((q_1, p_1, a_1)\) and \((q_2, p_2, a_2)\) such that (i) \(a_1\) is a named entity that is also mentioned in \(q_2\), (ii) \(a_2\) is not in \(q_1\), and (iii) \(p_1 \neq p_2\). The pairs can be combined to form a 2-hop question \((Q, \{p_1, p_2\}, a_2)\). To ensure the mentions \((a_1\) and its occurrence in \(q_2\) denoted as \(e_2\)) refer to the same entity, we check if: 1. Spacy entity tagger (Honnibal et al., 2020) tags \(a_1\) and \(e_2\) as entities of the same type. 2. A wikipedia search with \(a_1\) and \(e_2\) return identical 1st result.

S3. Filter Disconnected Single-Hop Pairs  We want connected 2-hop questions — questions that cannot be answered without using the answers of the constituent single-hop questions. The MuSiQue condition (2) states that for a 2-hop question to be connected, either sub-question \(q_i\) should not be correctly answered without its context \((M(q_i, \phi) \neq a_i)\) and the tail question \(q_2\) should not be correctly answered when \(a_1\) is removed from it \((M(q_2^{\text{m1}}, C) \neq a_2)\). Accordingly we use a two step filtering process to find connected 2-hop questions. For simplicity, and because the second condition already filters some tail questions, our current implementation enforces the first condition only on the head question, \(q_1\).

Filtering Head Nodes: We collect all questions that appear at least once as the head of composable 2-hop questions \((q_1)\) to create a set of head nodes. We create 5-fold train-test splits of this set and train two Longformer-Large models (different seeds) per split (train on three, validate and test on one). We generate answer predictions using the 2 models on their corresponding test splits resulting in 2 predictions per question. We accept a head question if its average answer F1 returns same result for \(a_1\) and \(e_2\).

Filtering Tail Nodes: We create a unique set of masked single-hop questions that occur as a tail node \((q_2)\) in any composable 2-hop questions. If the same single-hop question occurs in two 2-
hop questions with different masked entity, they both are added to the set. We combine the gold-paragraph with 9 distractor paragraphs (retrieved \(4\) using the question without the masked entities as query). Same as before, we create 5-fold train-test splits and use 2 Longformer-Large models to get 2 answer and support predictions per question. We accept a tail question if mean answer f1 \(< 0.25\) or if it’s \(< 0.75\) and mean support f1 \(< 1.0\).

Finally, only 2-hop questions for which both head and tail node are acceptable are kept. We call this process **Disconnection Filtering**.

**S4. Build Multihop Questions** We now have a set of connected 2-hop questions, which form directed edges of a graph. Any subset \(S\) of it can be used to create a connected multi-hop question. We use 6 types of reasoning graphs with 2-4 hops as shown in Table 1. To avoid very long questions, we limit single-hop questions to \(< 10\) tokens, the total length of questions in 2,3-hops to \(< 15\), and 3-hops to \(< 20\) tokens. To ensure diversity, we (1) cap the reuse of bridging entities and single-hop questions at 25 and 100 multi-hop questions respectively (2) remove any \(n\)-hop question that’s subset of any \(m\)-hop question \((m > n > 1)\).

**S5. Minimize Train-Test Leakage** We devise a procedure to create train, validation and test splits such that models cannot achieve high-scores via memorization enabled by train-test leakage, an issue observed in some existing datasets (Lewis et al., 2021). Our procedure ensures that the training set has no overlap with validation or the test sets, and tries to keep the overlap between validation and test sets minimal.

We consider two multi-hop questions \(Q_i\) and \(Q_j\) to overlap if any of the following are common between \(Q_i\) and \(Q_j\): (i) single-hop question (ii) answer to any single-hop question (iii) associated paragraph to any single-hop question. To minimize such overlap, we take a set of multi-hop questions, greedily find a subset of given size \((S)\) which least overlaps with its complement \((S')\), and then remove overlapping questions from \(S'\), to get train \((S)\) and dev+test set \((S')\). Then, we split dev+test to dev and test similarly. We ensure the distribution of source datasets of single-hop questions in train, dev and test are similar, and also control the proportion of 2-4 hop questions.

**S6. Build Contexts for Questions** For an \(n\)-hop question, the context has 20 paragraphs containing: (i) supporting paragraphs associated with its single-hop questions \(\{p_1, p_2 \ldots p_n\}\), (ii) distractor paragraphs retrieved using a query that is a concatenation of single-hop questions from which all intermediate answer mentions are removed. To make distractor paragraphs harder to identify, we retrieve them from the set of gold-paragraphs for the filtered single-hop question \((S1)\).

**S7. Crowdsource Question Compositions** Amazon MTurk crowdworkers\(^3\) composed short and coherent questions from our final question compositions (represented as DAGs), taking care to ensure that information from all single-hop questions is used, and answer to the composed question is same as the last single-hop question. During the entire process the workers can see associated paragraphs for each question. The crowdworkers also filtered out incorrectly composed DAGs where the bridged entity mentions do not refer to the same underlying entity. They were encouraged to write short questions, make implicit inferences when possible, and were allowed to split questions in two sentences if needed. We call the dataset at this stage **MuSiQue-Ans**.

**S8. Add Unanswerable Questions** For each answerable multi-hop RC instance we create a corresponding unanswerable multi-hop RC instance using the procedure similar to the one proposed in (Trivedi et al., 2020). For a multi-hop question we randomly sample any of its single-hop question and make it unanswerable by ensuring the answer to that single-hop question doesn’t appear in any of the paragraphs in context (except this requirement, the context is built as described in S6). Since one of the single-hop questions is unanswerable, the whole multi-hop question is unanswerable.

The task now is to predict whether the question is answerable, and predict the answer and support if it’s answerable. Given the questions for answerable and unanswerable pair are identical and the context marginally changes, models that rely on shortcuts find this new task very difficult. We call the dataset at this stage **MuSiQue-Full**.

\(^3\)17 of 100 workers passed a qualification test, and annotated the full dataset. They were paid 25, 40, 60 cents for each 2, 3, 4 hop question, amounting to \(\sim 15\) USD per hour. Total cost was \(\sim 11K\) USD.

---

\(^4\)We use BM25 algorithm via Elasticsearch for retrieval.
The statistics for MuSiQue-Ans (MuSiQue-Full has twice the number of questions in each cell) are shown in Table 2. MuSiQue constitutes unique 21020 single-hop questions, 4132 answers to multi-hop questions, 19841 answers to single-hop questions, and 7676 supporting paragraphs. MuSiQue has 6 types of reasoning graphs and 2-4 hops (Examples in Table 1).

The statistics for MuSiQue-Ans are of form (Q,C;A,P,s) where there’s an additional binary classification task to predict S, the answerability of Q based on C, also referred to as context sufficiency (Trivedi et al., 2020).

**Final Dataset** The statistics for MuSiQue-Ans (MuSiQue-Full has twice the number of questions in each cell) are shown in Table 2. MuSiQue constitutes unique 21020 single-hop questions, 4132 answers to multi-hop questions, 19841 answers to single-hop questions, and 7676 supporting paragraphs. MuSiQue has 6 types of reasoning graphs and 2-4 hops (Examples in Table 1).

## 6 Experimental Setup

### 6.1 Datasets

We compare our datasets (MuSiQue-Ans and MuSiQue-Full) with two similar multi-hop RC datasets: distractor-setting of HotpotQA (Yang et al., 2018) and 2WikiMultihopQA (Ho et al., 2020). Both datasets have 10 paragraphs as context. HQ and 2W have 2-hop and 2,4-hop questions respectively. Additionally, HQ has sentence support and 2W has entity-relation tuples support, but we don’t use this annotation in our training or evaluation for a fair comparison.

HQ, 2W, and Ans have 90K, 167K, and 20K training instances, respectively. For a fair comparison, we use equal sized training sets in all our experiments, obtained by randomly sampling 20K instances each from HQ and 2W, and referred to as HQ-20k and 2W-20k, respectively.

**Notation.** Instances in Ans, HQ, and 2W are of the form (Q,C;A,P,s). Given a question Q and context C consisting of a set of paragraphs, the task is to predict the answer A and identify supporting paragraphs P_s ∈ C. Ans additionally has gold decomposition G_Q (§3), which can be leveraged during training. Instances in Ans are of form (Q,C;A,P_s,S), where there’s an additional binary classification task to predict S, the answerability of Q based on C, also referred to as context sufficiency (Trivedi et al., 2020).

|       | 2-hop | 3-hop | 4-hop | Total (24,814) |
|-------|-------|-------|-------|----------------|
| Train | 14376 | 4387  | 1175  | 19938          |
| Dev   | 1252  | 760   | 405   | 2417           |
| Test  | 1271  | 763   | 425   | 2459           |

Table 2: Dataset statistics of MuSiQue-Ans. MuSiQue-Full contains twice the number of questions in each category above – one answerable and one unanswerable.

**Metrics.** For Ans, HQ, and 2W, we report the standard F1 based metrics for answer (An) and support identification (Sp); see Yang et al. (2018) for details. To make a fair comparison across datasets, we use only paragraph-level support F1.

For Ans, we follow Trivedi et al. (2020) to combine sufficiency prediction S with An and Sp, which are denoted as An+Sf and Sp+Sf. Instances in Ans/Full are evaluated in pairs. For each Q with a sufficient context C, there is a paired instance with Q and an insufficient context C′. For An+Sf, if a model incorrectly predicts context sufficiency (yes or no) for either of the instances in a pair, it gets 0 pts on that pair. Otherwise, it gets same An score on that pair as it gets on the answerable instance in that pair. Scores are averaged across all pairs of instances in the dataset. Likewise for Sp+Sf.

### 6.2 Models

Our models are Transformer-based (Vaswani et al., 2017) language models (Devlin et al., 2019), implemented using PyTorch (Paszke et al., 2019), HuggingFace Transformers (Wolf et al., 2019) and AllenNLP (Gardner et al., 2017). We experiment with 2 types of models: (1) Multi-hop Models, which are in principle capable of employing desired reasoning, and have demonstrated competitive performance on previous multi-hop QA datasets. They help probe the extent to which a dataset can be solved by current models. (2) Artifact-based Models, which are restricted in some way that prohibits them from doing desired reasoning (discussed shortly). They help probe the extent to which a dataset can be cheated. Next, we describe these models for Ans and -Full. For HQ and 2W, they work similar to Ans.

#### 6.2.1 Multi-hop Models

**End2End (EE) Model.** This model takes (Q,C) as input, runs it through a transformer, and predicts (A,P_s) as the output for Ans and (A,P_s,S) for -Full. We use Longformer-Large as it’s one of the few transformer architectures that is able to fit the full context, and follow Beltagy et al. (2020) for answer and support prediction. Answerability prediction is done via binary classification using CLS token.

Note that our Longformer EE model is a strong model for multi-hop reasoning. When trained on full datasets, its answer F1 is 78.4 (within 3 pts of published SOTA (Groeneveld et al., 2020a)) on HQ, and 87.7 (SOTA) on 2W.
Select+Answer (SA) Model. This model, inspired by Quark (Groeneveld et al., 2020b) and SAE (Tu et al., 2020), has two parts. First, a selector ranks and selects the $K$ most relevant paragraphs $C_K$ from $C$. Second, for MuSiQue-Ans, the answerer predicts the answer and supporting paragraphs based only on $C_K$. For MuSiQue-Full, it additionally predicts answerability. Both components are trained individually using annotations available in the dataset. We implement a selector using RoBERTa-large (Liu et al., 2019), and an answerer using Longformer-Large.

Step Execution (EX) Model. Similar to prior work (Talmor and Berant, 2018; Min et al., 2019b; Qi et al., 2020; Khot et al., 2021), this model performs explicit, step-by-step multi-hop reasoning, by first decomposing the $Q$ into a DAG $G_Q$ having single-hop questions, and then calling single-hop model repeatedly to execute this decomposition.

The decomposer is trained with gold decompositions, and is implemented with BART-large.

The executor takes $C$ and the predicted DAG $G_Q$, and outputs $(A, P_s)$ for MuSiQue-Ans and $(A, P_s, S)$ for MuSiQue-Full. It calls single-hop model $M_s$ repeatedly while traversing $G_Q$ along the edges and substituting the answers.

Model $M_s$ is trained on only single-hop instances—taking $(q, C)$ as input, and producing $(A, P_s)$ or $(A, P_{s_i}, S_i)$ as the output. Here $P_i$ refers to the supporting paragraph for $q_i$ and $S_i$ refers to whether $C$ is sufficient to answer $q_i$. For MuSiQue-Full, the answerer predicts $Q$ as having sufficient context if $M_s$ predicts all $q_i$ to have sufficient context. We implement 2 such single-hop models $M_s$: End2End and Select+Answer, abbreviated as EX(EE) and EX(SA) respectively.

We don’t experiment with this model on HQ, since it needs ground-truth decomposition and intermediate answers, which aren’t available in HQ.

6.2.2 Artifact-based Models

The Q-Only Model takes only $Q$ as input (no $C$) and generates output $A$ for $\exists$-Ans and $(A, S)$ for $\exists$-Full. We implement this with BART-large (Lewis et al., 2020a). The C-Only Model takes only $C$ as input (no $Q$) and predicts $(A, P_s)$ for $\exists$-Ans and $(A, P_s, S)$ for $\exists$-Full. We implement this with an EE Longformer-Large model with empty $Q$. The 1-Para Model, like Min et al. (2019a); Chen and Durrett (2019), is similar to SA model with $K=1$. Instead of training the selector to rank all $P_s$ the highest, we train it to rank any paragraph containing the answer $A$ as the highest. The answerer then takes as input one selected paragraph $p \in P_s$ and predicts an answer to $Q$ based solely on $p$. This model can’t access full supporting information, as all datasets have at least 2 supporting paragraphs.

Lastly, we find DiRe score of all datasets, which measures the extent to which the datasets can be cheated by Disconnected Reasoning (Trivedi et al., 2020). We report scores based on SA model since we found it to be strongest.

6.3 Human Performance

For a fair comparison across all the datasets, we find the human performance on HQ, 2W, and $\exists$-Ans using the same annotation guidelines and crowdsourcing setup. We sample 125 questions from each dataset, shuffle into a single set and obtain 5 annotations per question for both answer and supporting paragraphs.

We crowdsourced this task on Amazon Mechanical Turk. For the qualification round, we allowed only workers with master-qualification and selected workers who had more than 75 An and more than 75 Sp on all datasets. Only these 5 workers were allowed in rest of the annotation.

Unlike HotpotQA (Yang et al., 2018), where human scores were obtained by showing workers only supporting paragraphs, we show them all paragraphs for a fair comparison with models.

Using these annotations, we compute: 1. Human Majority (Maj) – most common annotated answer and support, and 2. Human Upper Bound (UB) – the answer and support that maximizes the score (strategy used in HotpotQA paper).

7 Empirical Findings

We now discuss our findings, demonstrating that MuSiQue is a challenging multi-hop dataset that is harder to cheat on than existing datasets (§7.1) and that the steps in the MuSiQue construction pipeline are individually valuable (§7.2). Finally, we explore avenues for future work (§7.3).

For HQ and 2W, we report numbers on the dev set. For $\exists$-Ans and $\exists$-Full, the only results in Table 4 are on test set, all others are on dev set.

---

7$K$ is a hyperparameter, chosen from $\{3, 5, 7\}$.

8https://www.mturk.com

9Our interface allowed for searching and sorting content easily with interactive text-overlap-based search queries.
7.1 MuSiQue is a Challenging Dataset

Compared to HQ and 2W, both variants of MuSiQue are less checkable via shortcuts and have a larger human-to-model gap.

| Human Models | HQ-20K | 2W-20K | -Ans |
|--------------|-------|-------|-------|
| Maj          | 83.6  | 92.0  | 84.1  |
| UB           | 91.8  | 96.0  | 98.9  |
| Multi-hop Models |       |       |       |
| EE           | 72.9  | 94.3  | 97.6  |
| SA           | 74.9  | 94.6  | 96.0  |
| EX(EE)       | —     | 79.8  | 97.5  |
| EX(SA)       | —     | 71.2  | 98.1  |
| Artifact Models |      |       |       |
| 1-Para       | 64.8  | —     | 32.0  |
| Q-only       | 19.6  | —     | 4.6   |
| DiRe Score   | 68.8  | 93.0  | 63.4  |

Table 3: Compared to other datasets, -Ans has a much higher human-model gap (top 2 rows), and is much less checkable (bottom 2 rows).

Higher Human-Model Gap. Top two sections of Table 3 show -Ans has a significantly higher human-model gap than the other datasets, for both answer and supporting paragraph identification. In fact, for both the other datasets, supporting paragraph identification has even surpassed human majority score, whereas for -Ans, there is 14 pts gap. Additionally, -Ans has a ~27 pt gap in answer f1, whereas HQ and 2W have a gap of only 10 and 5, respectively.

Our best model, EX(SA), scores 57.9, 47.9, and 28.1 answer f1 on 2, 3, and 4-hop questions of -Ans, resp. The EE model, on the other hand, stays around 42% irrespective of the number of hops.

Lower Cheatability. The 3rd section of Table 3 shows that the performance of artifact-based models (6.2.2) is much higher on HQ and 2W than on -Ans. E.g., the 1-para model achieves 64.8 and 60.1 answer score on HQ and 2W, resp., but only 32.0 on -Ans. Support identification in both datasets can be done to a surprisingly high degree (67.6 and 92.0 F1) even without the question (C-only model), but fails on -Ans.10

Similarly, the last row of Table 3 shows that the DiRe answer scores of HQ and 2W (68.8 and 63.4) are high, indicating that even disconnected reasoning can achieve such high scores. In contrast, this number is significantly lower (37.8) for -Ans.

These results demonstrate that -Ans is significantly less checkable via shortcut-based reasoning.

MuSiQue-Full: Even More Challenging. Table 4 shows that -Full is significantly more difficult and less checkable than -Ans.

Table 4: -Full is harder (top row) and less checkable (bottom row) than -Ans. Note: -Full has a stricter metric that operates over instance pairs (§6.1:metrics).

Intuitively, because the answerable and unanswerable instances are very similar but have different labels, it’s difficult for models to do well on both instances if they learn to rely on shortcuts (Kaushik et al., 2019; Gardner et al., 2020). All artifact-based models barely get any An+Sp or Sp+Sf score. For all multi-hop models too, the An drops by 14-17 pts and Sp by 33-44 pts.

7.2 Dataset Construction Steps are Valuable

Next, we show that the key steps of our dataset construction pipeline (§5) are valuable.

Disconnection Filter (step 3). To assess the effect of Disconnection Filter (DF), we ablate it from the pipeline, i.e., skip the filtering composite 2-hop questions to connected 2-hop questions. As we don’t have human-generated composed questions for the resulting questions, we use a seq2seq BART-large model that’s trained (using MuSiQue) to compose questions from input decomposition DAG. For a fair comparison, we randomly subsample train set from ablated pipeline to be of the same size as the original train set.

Table 5 shows that DF is crucial for increasing difficulty and reducing cheatability of the dataset. Without DF, both multi-hop and artifact-based models do much better on the resulting datasets.

10Even when -Ans is modified to have 10 paragraphs like HQ, C-only support score remains low; cf. Table 6.
Reduced Train-Test Leakage (step 5). To assess the effect of Reduced train-test Leakage (RL), we create a dataset the traditional way, with a random partition into train, validation, and test splits. For uniformity, we ensure the distribution of 2-4 hop questions in development set of the resulting dataset from both ablated pipelines remains the same as in the original development set. Like DF ablation, we also normalize train set sizes.

Table 5 shows that without a careful split, the dataset is highly solvable by multi-hop models (An=87.3). Importantly, most of this high score can also be achieved by artifact-based models: 1-para (An=85.1) and C-only (An=69.5), revealing the high cheatability of such a split.

Harder Distractors (step 7). To assess the effect of distractors in -Ans, we create 4 variations. Two vary the number of distractors: (i) 10 paragraphs and (ii) 20 paragraphs; and two vary the source: (i) Full wikipedia (FW)\(^{11}\) and (ii) gold context paragraphs from the good single-hop questions from step 1. We refer to the last setting as positive distractors (PD), as these paragraphs are likely to appear as supporting (positive) paragraphs in our final dataset.

| Ctxt Corpus | 1-Para | C-Only | EE |
|-------------|--------|--------|----|
|             | An | Sp | An | Sp | An | Sp |
| 10 FW        | 42.5 | — | 12.5 | 77.7 | 57.2 | 87.6 |
| 10 PD        | 28.0 | — | 5.5 | 34.6 | 54.1 | 80.2 |
| 20 FW        | 41.7 | — | 12.4 | 66.4 | 50.3 | 80.8 |
| 20 PD        | 32.0 | — | 3.4 | 0.0 | 42.3 | 67.6 |

Table 6: Positive Distractors (PD) are more effective than using Full Wikipedia (FW) for choosing distractors, as shown by lower scores of models. The effect of using PD is more pronounced when combined with the use of 20 (rather than 10) distractor paragraphs.

Table 6 shows that all models find PD signif-

\(^{12}\)Our single-hop datasets are wikipedia-based, and we ensured retrieved contexts from FW are 20-300 words, like PD.

\(^{13}\)SA worked best for 7 selected paragraphs, where the answerer (T5) had to process \(\sim 1100\) wordpieces on average.
to use such a carefully controlled process to create a challenging dataset that, by design, requires connected reasoning by reducing potential reasoning shortcuts, minimizing train-test leakage, and including harder distractor contexts. Empirical results show that Ans has a substantially higher human-model gap and is significantly less cheatable via disconnected reasoning than previous datasets. The dataset also comes with unanswerable questions, and question decompositions which we hope spurs further work in developing models that get right answers for the right reasons.

References

Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. arXiv:2004.05150.

Jifan Chen and Greg Durrett. 2019. Understanding dataset design choices for multi-hop reasoning. In NAACL-HLT.

Wenhu Chen, Hanwen Zha, Zhiyu Chen, Wenhan Xiong, Hong Wang, and William Wang. 2020. Hybridqa: A dataset of multi-hop question answering over tabular and textual data. Findings of EMNLP 2020.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In NAACL.

Dheeru Dua, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. 2019. DROP: A reading comprehension benchmark requiring discrete reasoning over paragraphs. In NAACL.

Hady ElSahar, P. Vougiouklis, Arslan Remaci, C. Gravier, Jonathon S. Hare, F. Laforest, and E. Simperl. 2018. T-REx: A large scale alignment of natural language with knowledge base triples. In LREC.

James Ferguson, Matt Gardner, Hannaneh Hajishirzi, Tushar Khot, and Pradeep Dasigi. 2020. IIRC: A dataset of incomplete information reading comprehension questions. In EMNLP.

Matt Gardner, Yoav Artzi, Victoria Basmova, Jonathan Berant, Ben Bogin, Sihao Chen, Pradeep Dasigi, Dheeru Dua, Yanai Elazar, Ananth Gottumukkala, Nitish Gupta, Hanna Hajishirzi, Gabriel Ilharco, Daniel Khastahi, Kevin Lin, Jiachong Liu, Nelson F. Liu, Phoebe Mulcaire, Qiang Ning, Sameer Singh, Noah A. Smith, Sanjay Subramanian, Reut Tsarfaty, Eric Wallace, A. Zhang, and Ben Zhou. 2020. Evaluating models’ local decision boundaries via contrast sets. In Findings of EMNLP.

Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson F. Liu, Matthew Peters, Michael Schmitz, and Luke S. Zettlemoyer. 2017. AllenNLP: A deep semantic natural language processing platform. arXiv preprint arXiv:1803.07640.

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. TACL.

Dirk Groeneveld, Tushar Khot, Mausam, and Ashish Sabharwal. 2020a. A simple yet strong pipeline for HotpotQA. In EMNLP.

Dirk Groeneveld, Tushar Khot, Ashish Sabharwal, et al. 2020b. A simple yet strong pipeline for hotpotqa. In EMNLP.

Xanh Ho, A. Nguyen, Saku Sugawara, and Akiko Aizawa. 2020. Constructing a multi-hop qa dataset for comprehensive evaluation of reasoning steps. In COLING.

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. SpaCy: Industrial-strength natural language processing in Python.

Peter Jansen, Elizabeth Wainwright, Steven Marmorstein, and Clayton Morrison. 2018. WorldTree: A corpus of explanation graphs for elementary science questions supporting multi-hop inference. In Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018), Miyazaki, Japan. European Language Resources Association (ELRA).

Yichen Jiang, Shikha Bordia, Zheng Zhong, Charles Dognin, Maneesh Singh, and Mohit Bansal. 2020. HoVer: A dataset for many-hop
fact extraction and claim verification. In *Findings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Divyansh Kaushik, Eduard Hovy, and Zachary Lipton. 2019. Learning the difference that makes a difference with counterfactually-augmented data. In *ICLR*.

Divyansh Kaushik and Zachary C. Lipton. 2018. How much reading does reading comprehension require? a critical investigation of popular benchmarks. In *EMNLP*.

D. Khashabi, S. Min, T. Khot, A. Sabhwaral, O. Tafjord, P. Clark, and H. Hajishirzi. 2020. Unifiedqa: Crossing format boundaries with a single qa system. *EMNLP - findings*.

Daniel Khaseh, Snigdha Chaturvedi, Michael Roth, Shyam Upadhyay, and Dan Roth. 2018. Looking beyond the surface:a challenge set for reading comprehension over multiple sentences. In *NAACL*.

Tushar Khot, Peter Clark, Michal Guerquin, Peter Jansen, and Ashish Sabharwal. 2020. QASC: A dataset for question answering via sentence composition. In *AAAI*.

Tushar Khot, Daniel Khaseh, Kyle Richardson, Peter Clark, and Ashish Sabharwal. 2021. Text modular networks: Learning to decompose tasks in the language of existing models. In *NAACL*.

T. Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur P. Parikh, Chris Alberti, D. Epstein, Illia Polosukhin, J. Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc V. Le, and Slav Petrov. 2019. Natural questions: A benchmark for question answering research. *TACL*, 7:453–466.

Omer Levy, Minjoon Seo, Eunsol Choi, and Luke Zettlemoyer. 2017. Zero-shot relation extraction via reading comprehension. In *CoNLL*.

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 gen-eration, translation, and comprehension. In *ACL*.

Patrick Lewis, Barlas Oğuz, Ruty Rinott, Sebastian Riedel, and Holger Schwenk. 2020b. MLQA: Evaluating cross-lingual extractive question answering. In *ACL*.

Patrick Lewis, Pontus Stenetorp, and Sebastian Riedel. 2021. Question and answer test-train overlap in open-domain question answering datasets. In *EACL*.

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. *arXiv preprint arXiv:1907.11692*.

Sewon Min, Eric Wallace, Sameer Singh, Matt Gardner, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2019a. Compositional questions do not necessitate multi-hop reasoning. In *ACL*.

Sewon Min, Victor Zhong, Luke S. Zettlemoyer, and Hannaneh Hajishirzi. 2019b. Multi-hop reading comprehension through question decomposition and rescoring. In *ACL*.

Liangming Pan, Wenhu Chen, Wenhan Xiong, Min-Yen Kan, and William Yang Wang. 2021. Unsupervised multi-hop question answering by question generation. In *NAACL*.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. PyTorch: An imperative style, high-performance deep learning library. In *NeurIPS*, pages 8024–8035.

Fabio Petroni, Aleksandra Piktus, Angela Fan, Patrick Lewis, Majid Yazdani, Nicola De Cao, James Thorne, Yacine Jernite, Vassilis Plachouras, Tim Rocktäschel, and Sebastian Riedel. 2021. KILT: A benchmark for knowledge intensive language tasks. In *NAACL*.

Peng Qi, Haejun Lee, Oghenetegiri Sido, Christopher D Manning, et al. 2020. Retrieve, read,
rerank, then iterate: Answering open-domain questions of varying reasoning steps from text. *arXiv preprint arXiv:2010.12527.*

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don’t know: Unanswerable questions for SQuAD. In *ACL.*

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100, 000+ questions for machine comprehension of text. In *EMNLP.*

Alon Talmor and Jonathan Berant. 2018. The web as a knowledge-base for answering complex questions. In *NAACL.*

Alon Talmor, Ori Yoran, Amnon Catav, Dan Lahav, Yizhong Wang, Akari Asai, Gabriel Ilharco, Hannaneh Hajishirzi, and Jonathan Berant. 2021. MultiModalQA: Complex question answering over text, tables and images. In *ICLR.*

H. Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2020. Is multihop QA in DiRe condition? Measuring and reducing disconnected reasoning. In *EMNLP.*

Ming Tu, Kevin Huang, Guangtao Wang, Jing Huang, Xiaodong He, and Bowen Zhou. 2020. Select, answer and explain: Interpretable multihop reading comprehension over multiple documents. In *AAAI.*

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NeurIPS,* pages 5998–6008.

Johannes Welbl, Pontus Stenetorp, and Sebastian Riedel. 2018. Constructing datasets for multihop reading comprehension across documents. *TACL,* 6:287–302.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, R’emi Louf, Morgan Funtowicz, and Jamie Brew. 2019. Huggingface’s transformers: State-of-the-art natural language processing. *ArXiv,* abs/1910.03771.

Tomer Wolfson, Mor Geva, Ankit Gupta, Matt Gardner, Yoav Goldberg, Daniel Deutch, and Jonathan Berant. 2020. Break it down: A question understanding benchmark. *TACL.*

Ledell Wu, Fabio Petroni, Martin Josifoski, Sebastian Riedel, and Luke Zettlemoyer. 2020. Zero-shot entity linking with dense entity retrieval. In *EMNLP.*

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. HotpotQA: A dataset for diverse, explainable multihop question answering. In *EMNLP.*

Ori Yoran, Alon Talmor, and Jonathan Berant. 2021. Turning tables: Generating examples from semi-structured tables for endowing language models with reasoning skills. *arXiv preprint arXiv:2107.07261.*
A Appendix

Table 7 shows Examples of 6 different reasoning graph shapes in MuSiQue-Ans. Figure 3 shows our annotation interface for question composition. Figure 4 shows our annotation interface for establishing human scores on MuSiQue-Ans, 2Wiki-MultihopQA and HotpotQA.
Figure 3: This is the annotation interface we used for MuSiQue. Workers could see decomposition graph and passage associated with subquestions.
Figure 4: This is the annotation interface we used for comparing human-scores on MuSiQue, HotpotQA and 2WikiMultihopQA.
Who succeeded the first President of Namibia? Hifikepunye Pohamba

1. Who was the first President of Namibia? Sam Nujoma
2. Who succeeded Sam Nujoma? Hifikepunye Pohamba

What currency is used where Billy Giles died? pound sterling

1. At what location did Billy Giles die? Belfast
2. What part of the United Kingdom is Belfast located in? Northern Ireland
3. What is the unit of currency in Northern Ireland? pound sterling

1. Billy Giles (..., Belfast – 25 September 1998, Belfast)
2. ... thirty-six public houses ... Belfast, Northern Ireland.
3. bank for pound sterling, issuing ...
   in Northern Ireland.

When was the first establishment that McDonaldization is named after, open in the country Horndean is located? 1974

1. What is McDonaldization named after? McDonald’s
2. Which state is Horndean located in? England
3. When did the first McDonald’s open in England? 1974

1. ... spreading of McDonald’s restaurants ...
2. ... Horndean is a village ... in Hampshire, England.
3. 1974 ... first McDonald’s in the United Kingdom .. in London.

When did Napoleon occupy the city where the mother of the woman who brought Louis XVI style to the court died? 1805

1. Who brought Louis XVI style to the court? Marie Antoinette
2. Who’s mother of Marie Antoinette? Maria Theresa
3. In what city did Maria Theresa die? Vienna
4. When did Napoleon occupy Vienna? 1805

1. Marie Antoinette, ...brought the "Louis XVI" style to court
2. Maria Antonia of Austria, youngest daughter of .. Maria Theresa
3. Maria Theresa ... in Vienna ... after the death
4. occupation of Vienna by Napoleon’s troops in 1805

How many Germans live in the colonial holding in Aruba’s continent that was governed by Prazeres’s country? 5 million

1. What continent is Aruba in? South America
2. What country is Prazeres? Portugal
3. The colonial holding in South America governed by Portugal? Brazil
4. How many Germans live in Brazil? 5 million

1. ... Aruba, lived including indigenous peoples of South America
2. Prazeres is .. in municipality of Lisbon, Portugal.
3. Portugal, ... desire for independence amongst Brazilians.
4. Brazil ... 5 million people claiming German ancestry.

When did the people who captured Malakoff come to the region where Philipsburg is located? 1625

1. What is Philipsburg capital of? Saint Martin
2. Saint Martin (French part) is located on what terrain feature? Caribbean
3. Who captured Malakoff? French
4. When did the French come to the Caribbean? 1625

1. Philipsburg .. capital of .. Saint Martin
2. ... airport on the Caribbean island of Saint Martin/Sint Maarten.
3. ... the capture of the Malakoff by the French
4. French trader ... sailed to Carribean in 1625 ... French settlement on

Table 7: Examples of 6 different reasoning graph shapes in MuSiQue-Ans