Is Multihop QA in DiRe Condition? Measuring and Reducing Disconnected Reasoning

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

Has there been real progress in multi-hop question-answering? Models often exploit dataset artifacts to produce correct answers, without connecting information across multiple supporting facts. This limits our ability to measure true progress and defeats the purpose of building multi-hop QA datasets. We make three contributions towards addressing this. First, we formalize such undesirable behavior as disconnected reasoning across subsets of supporting facts. This allows developing a model-agnostic probe for measuring how much any model can cheat via disconnected reasoning. Second, using a notion of *contrastive support sufficiency*, we introduce an automatic transformation of existing datasets that reduces the amount of disconnected reasoning. Third, our experiments[1] suggest that there hasn't been much progress in multifact QA in the reading comprehension setting. For a recent large-scale model (XLNet), we show that only 18 points out of its answer F1 score of 72 on HotpotQA are obtained through multifact reasoning, roughly the same as that of a simpler RNN baseline. Our transformation substantially reduces disconnected reasoning (19 points in answer F1). It is complementary to adversarial approaches, yielding further reductions in conjunction.

1 Introduction

Multi-hop question answering requires connecting and synthesizing information from multiple facts in the input text, a process we refer to as *multifact reasoning*. Prior work has, however, shown that bad reasoning models, ones that by design do not connect information from multiple facts, can achieve high scores because they can exploit specific types of biases and artifacts (e.g., answer type shortcuts) in existing datasets (Min et al., 2019; Chen and Durrett, 2019). While this demonstrates the existence of models that *can* cheat, what we do not know is the extent to which current models *do* cheat, and whether there has been real progress in building models for multifact reasoning.

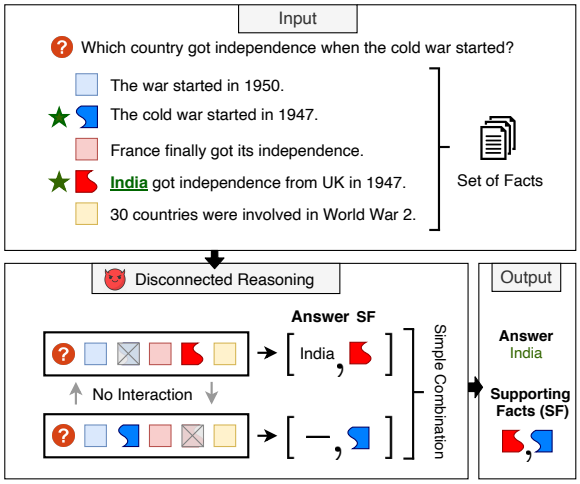

Figure 1: Example of *disconnected reasoning*, a form of bad multifact reasoning: Model arrives at the answer by simply combining its outputs from two subsets of the input, neither of which contains all supporting facts. From one subset, it identifies the blue supporting fact ( ), the only one mentioning cold war. *Independently*, from the other subset, it finds the red fact ( ) as the only one mentioning a country getting independence with associated time, and returns the correct answer (India). Further, it returns a simple union of the supporting facts it found over the input subsets.

We address this issue in the context of multi-hop reading comprehension. We introduce a general-purpose characterization of a form of bad multihop reasoning, namely *disconnected reasoning*. For datasets annotated with supporting facts, this allows devising a model-agnostic probe to estimate the extent of disconnected reasoning done by *any* model, and an automatic transformation of existing datasets that reduces such disconnected reasoning.

[*] Early portion of this work was done during the first author's internship at Allen Institute for AI.

[1] https://github.com/stonybrooknlp/dire

Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, pages 8846–8863,
November 16–20, 2020.

Measuring Disconnected Reasoning. Good multifact reasoning,[2] at a minimum, requires models to connect information from one or more facts when they select and use information from other facts to arrive at an answer. However, models can cheat, as illustrated in Figure 1, by independently assessing information in subsets of the input facts none of which contains all supporting facts, and taking a simple combination of outputs from these subsets (e.g., by taking a union) to produce the overall output. This entirely avoids meaningfully combining information across all supporting facts, a fundamental requirement of multifact reasoning. We refer to this type of reasoning as *disconnected reasoning* (DiRe in short) and provide a formal criterion, the DIRE condition, to catch cheating models. Informally, it checks whether for a given test of multifact reasoning (e.g., answer prediction or supporting fact identification), a model is able to trivially combine its outputs on subsets of the input context (none of which has all supporting facts) without any interaction between them.

Using the DIRE condition, we develop a systematic probe, involving an automatically generated probing dataset, that measures how much a model can score using disconnected reasoning.

Reducing Disconnected Reasoning. A key aspect of a disconnected reasoning model is that it does not change its behavior towards the selection and use of supporting facts that are in the input, whether or not the input contains all of the supporting facts the question requires. This suggests that the notion of sufficiency—whether all supporting facts are present in the input, which clearly matters to a good multifact model—does not matter to a bad model. We formalize this into a *constrastive support sufficiency* test (CSST) as an additional test of multifact reasoning that is harder to cheat. We introduce an automatic transformation that adds to each question in an original multi-hop dataset a group of *insufficient context* instances corresponding to different subsets of supporting facts. A model must recognize these as having insufficient context in order to receive any credit for the question.

Our **empirical evaluation** on the HotpotQA dataset (Yang et al., 2018) reveals three interesting findings: (i) A substantial amount of progress on multi-hop reading comprehension can be attributed to improvements in disconnected reasoning. E.g., XLNet (Yang et al., 2019), a recent large-scale language model, only achieves 17.5 F1 pts (of its total 71.9 answer F1) via multifact reasoning, roughly the same as a much simpler RNN model. (ii) Training on the transformed dataset with CSST results in a substantial reduction in disconnected reasoning (e.g., a 19 point drop in answer F1), demonstrating that it less cheatable, is a harder test of multifact reasoning, and gives a better picture of the current state of multifact reasoning. (iii) The transformed dataset is more effective at reducing disconnected reasoning than a previous adversarial augmentation method (Jiang and Bansal, 2019), and is also complementary, improving further in combination.

In summary, the DiRe probe serves as a simple yet effective tool for **model designers** to assess how much of their model's score can actually be attributed to multifact reasoning. Similarly, **dataset designers** can assess how cheatable is their dataset D (in terms of allowing disconnected reasoning) by training a strong model on the DiRe probe for D, and use our transform to reduce D's cheatability.

2 Related Work

Multi-hop Reasoning: Many multifact reasoning approaches have been proposed for HotpotQA and similar datasets (Mihaylov et al., 2018; Khot et al., 2020). These use iterative fact selection (Nishida et al., 2019; Tu et al., 2020; Asai et al., 2020; Das et al., 2019), graph neural networks (Xiao et al., 2019; Fang et al., 2020; Tu et al., 2020), or simply cross-document self-attention (Yang et al., 2019; Beltagy et al., 2020) to capture inter-paragraph interaction. While these approaches have pushed the state of the art, the extent of actual progress on multifact reasoning remains unclear.

Identifying Dataset Artifacts: Several works have identified dataset artifacts for tasks such as NLI (Gururangan et al., 2018), Reading Comprehension (Feng et al., 2018; Sugawara et al., 2020), and even multi-hop reasoning (Min et al., 2019; Chen and Durrett, 2019). These artifacts allow models to solve the dataset without actually solving the underlying task. On HotpotQA, prior work has shown *existence* of models that identify the support (Groeneveld et al., 2020) and answer (Min et al., 2019; Chen and Durrett, 2019) by operating on each paragraph or sentence independently. We, on the other hand, estimate the amount of disconnected reasoning in *any* model and quantify the cheatability of answer and support identification.

[2] We refer to desirable types of multifact reasoning as *good* and undesirable types as *bad*.

Mitigation of Dataset Artifacts: To deal with these artifacts, several adversarial methods have been proposed for reading comprehension (Jia and Liang, 2017; Rajpurkar et al., 2018) and multi-hop QA (Jiang and Bansal, 2019). These methods minimally perturb the input text to limit the effectiveness of the dataset artifacts. Our insufficient context instances that partition the context are complementary to these approaches (as we show in our experiments). Rajpurkar et al. (2018) used a mix of answerable and unanswerable questions to make the models avoid superficial reasoning. In a way, while these hand-authored unanswerable questions also provide insufficient context, we specifically focus on (automatically) creating unanswerable *multi-hop questions* by providing insufficient context.

Minimal Pairs: Recent works (Kaushik et al., 2019; Lin et al., 2019; Gardner et al., 2020) have proposed evaluating NLP systems by generating minimal pairs (or contrastive examples) that are similar but have different labels. Insufficient context instances in our sufficiency test can be thought of as automatically generated contrastive examples specifically for avoiding disconnected reasoning.

3 Measuring Disconnected Reasoning

This section formalizes the DIRE *condition*, which captures what it means for a model to employ disconnected reasoning, and describes how to use this condition to probe the amount of disconnected reasoning performed by a given model, and the extent of such reasoning possible on a dataset.

A good multifact reasoning is one where information from all the supporting facts is meaningfully synthesized to arrive at an answer. The precise definition for what constitutes meaningful synthesis is somewhat subjective; it depends on the semantics of the facts and the specific question at hand, making it challenging to devise a measurable test for the amount of multifact (or non-multifact) reasoning done by a model or needed by a dataset.

Previous works have used the Answer Prediction task (i.e., identifying the correct answer) and the Supporting Fact Identification task (identifying all facts supporting the answer) as approximate *tests* of multifact reasoning. We argue that, *at a minimum*, good multifact reasoning requires connected reasoning—one where information from at least one supporting fact is connected to the selection and use of information from other supporting facts. Consider the example question in Figure 1. A good multifact reasoning will look for a supporting fact that mentions when the cold war started ( ) and use information from this fact (year 1947) to select the other supporting fact mentioning the country that got independence ( ) (or vice versa).

A bad multifact reasoning model, however, can cheat on answer prediction by only looking for a fact that mentions a country getting independence at some time (mentioned in), without connecting this to when the cold war started (mentioned in). Similarly, the model can also cheat on supporting fact identification by treating it as two independent sub-tasks—one returning a fact mentioning the time when a country got independence, and another for a fact mentioning the time when cold war started. The result of at least one of the two sub-tasks should influence the result of the other sub-task, but here it does not. This results in *disconnected reasoning*, where *both* supporting facts are identified without reference to the other.[3] Even though the precise definition of a meaningful synthesis of information is unclear, it is clear that models performing this type of disconnected reasoning cannot be considered as doing valid multifact reasoning. Neither answer prediction nor support identification directly checks for such disconnected reasoning.

3.1 Disconnected Reasoning

We can formalize the notion of disconnected reasoning from the perspective of any multihop reasoning test. For the rest of this work, we assume a multifact reading comprehension setting, where we have a dataset D with instances of the form $q = (Q, C; A)$. Given a question Q along with a context C consisting of a set of facts, the task is to predict the answer A. C includes a subset F_s of at least two facts that together provide support for A.

Let τ denote a test of multifact reasoning and $\tau(q)$ the output a model should produce when tested on input q. Consider the Support Identification test, where $\tau(q) = F_s$. Let there be two proper subsets of the supporting facts F_{s1} and F_{s2} such that $F_s = F_{s1} \cup F_{s2}$. We argued above that a model performs disconnected reasoning if it does not use information in F_{s1} to select and use information in F_{s2} and vice versa. One way we can catch this behavior is by checking if the model is able to identify F_{s1} given $C \setminus F_{s2}$ and identify F_{s2}

[3] Identifying one of the facts in isolation is fine, as long as information from this fact is used to identify the other fact.

given $C \setminus F_{s1}$. To pass the test successfully the model only needs to trivially combine its outputs from the two subsets $C \setminus F_{s2}$ and $C \setminus F_{s1}$.

Concretely, we say M performs Disconnected Reasoning on q from the perspective of a test τ if the following condition holds:

> **DiRe condition**: There exists a proper bi-partition[4] $\{F_{s1}, F_{s2}\}$ of F_s such that the two outputs of M with input q modified to have $C \setminus F_{s2}$ and $C \setminus F_{s1}$ as contexts, respectively, can be *trivially combined* to produce $\tau(q)$.

The need for considering all proper bi-partitions is further explained in Appendix A.2, using an example of 3-hop reasoning (Figure 8). The DiRe condition does not explicitly state what constitutes a trivial combination; this is defined below individually for each test. We note that it only captures disconnected reasoning, which is one manifestation of the lack of a meaningful synthesis of facts.

For Answer Prediction, trivial combination corresponds to producing answers (which we assume are associated confidence scores) independently for the two contexts, and choosing the answer with the highest score. Suppose M answers a_1 with score $s(a_1)$ on the first context in the DiRe condition; similarly a_2 for the second context. We say the condition is met if $A = \arg\max_{a \in \{a_1, a_2\}} s(a)$.

For the Support Identification test, as in the example discussed earlier, *set union* constitutes an effective trivial combination. Suppose M identifies G_1 and G_2 as the sets of supporting facts for the two inputs in the DiRe condition, respectively. We say the condition is met if $G_1 \cup G_2 = F_s$.

In the above discussion, we assumed the so called 'exact match' or EM metric for assessing whether the answer or supporting facts produced by the combination operator were correct. In general, let $m_\tau(q, \mu(q))$ be any metric for scoring the output $\mu(q)$ of a model against the true label $\tau(q)$ for a test τ on question q (e.g., answer EM, support F1, etc.). We can apply the same metric to the output of the combination operator (instead of $\mu(q)$) to assess the extent to which the DiRe condition is met for q under the metric m_τ.

3.2 Probing Disconnected Reasoning

The DiRe condition allows devising a probe for measuring how much can a model cheat on a test τ, i.e., how much can it score using disconnected reasoning. The probe for a dataset D comprises an *automatically generated* dataset $\mathbb{P}_\tau(D)$, on which the model is evaluated, with or without training.

For simplicity, consider the case where $F_s = \{f_1, f_2\}$. Here $\{\{f_1\}, \{f_2\}\}$ is the unique proper bi-partition of F_s. The DiRe condition checks whether a model M can arrive at the correct test output $\tau(q)$ for input $q = (Q, C; A)$ by trivially combining its outputs on contexts $C \setminus \{f_1\}$ and $C \setminus \{f_2\}$. Accordingly, for each $q \in D$, the probing dataset $\mathbb{P}_{\text{ans+supp}}(D)$ for Answer Prediction and Support Identification contains a *group of instances*:

$$(Q, C \setminus \{f_1\}; L^{?}_{\text{ans}}{=}A, L_{\text{supp}}{=}\{f_2\}) \quad (1)$$

$$(Q, C \setminus \{f_2\}; L^{?}_{\text{ans}}{=}A, L_{\text{supp}}{=}\{f_1\}) \quad (2)$$

where L_{supp} denotes the support identification label and $L^{?}_{\text{ans}}{=}A$ represents an optional answer label that is included only if A is present in the supporting facts retained in the context. These labels are only used if the model is trained on $\mathbb{P}_\tau(D)$.

Models operate independently over instances in $\mathbb{P}_\tau(D)$, whether or not they belong to a group. Probe performance, however, is measured via a **grouped probe metric**, denoted $m^{\mathbb{P}}_\tau$, that captures how well does the *trivial combination* of the two corresponding outputs match $\tau(q)$ according to metric m_τ, as per the DiRe condition for τ. Specifically, for Answer Prediction, we use the highest scoring answer (following the *argmax* operator in Section 3.1) across the two instances in the group, and evaluate it against A using a standard metric m_{ans} (EM, F1, etc.). For Support Identification, we take the *union* of the two sets of supporting facts identified (for the two instances), and evaluate it against $\{f_1, f_2\}$ using a standard metric m_{supp}.

General case of $|F_s| \geq 2$: We translate each $q \in D$ into a *collection* $\mathbb{P}_\tau(q)$ of $2^{|F_s|-1} - 1$ groups of instances, with $\mathbb{P}_\tau(q; s_1)$ denoting the group for the bi-partition $\{F_{s1}, F_{s2}\}$ of F_s. This group, for answer and support prediction tests, contains:

$$(Q, C \setminus F_{s1}; L^{?}_{\text{ans}}{=}A, L_{\text{supp}}{=}F_{s2}) \quad (3)$$

$$(Q, C \setminus F_{s2}; L^{?}_{\text{ans}}{=}A, L_{\text{supp}}{=}F_{s1}) \quad (4)$$

As per the DiRe condition, as long as the model cheats on any one bi-partition, it is considered to cheat on the test. Accordingly, the probe metric $m^{\mathbb{P}}_\tau$ uses a *disjunction* over the groups:

$$m^{\mathbb{P}}_\tau(q, \mu(\mathbb{P}_\tau(q))) = \max_{\{F_{s1}, \_\}} m^{\mathbb{P}}_\tau(q, \mu(\mathbb{P}_\tau(q; s_1))) \quad (5)$$

[4] $\{X, Y\}$ is a proper bi-partition of a set Z if $X \cup Y = Z, X \cap Y = \phi, X \neq \phi$, and $Y \neq \phi$.

where $\mu(\mathbb{P}_\tau(q))$ denotes the model's prediction on the probe group $\mathbb{P}_\tau(q)$, the max is over all proper bi-partitions of F_s, and $m^{\mathbb{P}}_\tau(q, \mu(\mathbb{P}_\tau(q; s_1)))$ denotes the probe metric for the group $\mathbb{P}_\tau(q; s_1)$ which incorporates the trivial combination operator, denoted $\oplus$, associated with τ as follows:

$$m^{\mathbb{P}}_\tau(q, \mu(\mathbb{P}_\tau(q; s_1))) = m_\tau(q, \oplus_{q' \in \mathbb{P}(q;s_1)} \mu(q')) \quad (6)$$

For example, when τ is answer prediction, we view $\mu(q')$ as both the predicted answer and its score for q', $\oplus$ chooses the answer with the highest score, and m_{ans} evaluates it against A. When τ is support identification, $\mu(q')$ is the set of facts the model outputs for q', $\oplus$ is union, and m_{supp} is a standard evaluation of the result against the true label F_s.

3.3 Use Cases of DiRe Probe

The probing dataset $\mathbb{P}_\tau(D)$ can be used by **model designers** to assess what portion of their model M's performance on D can be achieved on a test τ via disconnected reasoning, by computing:

$$\text{DiRe}^\tau(M, D) = \text{S}^\tau_{\text{cond}}(M, \mathbb{P}_\tau(D) \mid D) \quad (7)$$

This is a zero-shot evaluation[5] where M is not trained on $\mathbb{P}_\tau(D)$. $\text{S}^\tau_{\text{cond}}$ represents M's score on $\mathbb{P}_\tau(D)$ conditioned on its score on D, computed as the question-wise minimum of M's score on D and $\mathbb{P}_\tau(D)$, in terms of metrics m_τ and $m^{\mathbb{P}}_\tau$, resp.[6]

Similarly, $\mathbb{P}_\tau(D)$ can be used by a **dataset designer** to assess how cheatable D is by computing:

$$\text{DiRe}^\tau(D) = \text{S}^\tau(M^*, \mathbb{P}_\tau(D)) \quad (8)$$

where M^* is the strongest available model architecture for τ that is trained on $\mathbb{P}_\tau(D)$ and S^τ is its score under metric $m^{\mathbb{P}}_\tau$.

4 Reducing Disconnected Reasoning

This section introduces an automatic transformation of a dataset to make it less cheatable by disconnected reasoning. It also defines a probe dataset for assessing how cheatable the transformed dataset is.

A disconnected reasoning model does not connect information across supporting facts. This has an important consequence: when a supporting fact is dropped from the context, the model's behavior on other supporting facts remains unchanged. Figure 2 illustrates this for the example in Figure 1.

[5] Our experiments also include inoculation (i.e., finetuning on a small fraction of the dataset) before evaluation.

[6] For answer prediction with exact-match, this corresponds to M getting 1 point for correctly answering a question group in $\mathbb{P}(\mathcal{D})$ only if it correctly answers the corresponding original question in $\mathcal{D}$ as well.

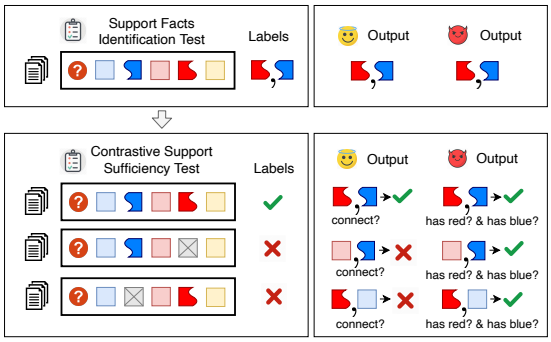

Figure 2: Transformation of a question for Contrastive Support Sufficiency evaluation. Top-Left: Original instance q labeled with red ( ) and blue ( ) supporting facts F_s. Bottom-Left: Its transformation into a group $\mathbb{T}(q)$ of 3 instances, one with sufficient and two with insufficient context, with labels denoting context sufficiency. Right: Behavior of good vs. bad models on q and $\mathbb{T}(q)$. A good multifact model would realize that the potentially relevant facts are not sufficient (do not connect) whereas a bad model would find potentially relevant facts and assume they are sufficient.

Suppose we create an insufficient context C' by removing the blue fact ( ) from C (shown in the last row of Figure 2, with the removed fact crossed out). With the full context C, the cheating model discussed earlier did not use the information in the blue fact ( ) to produce the answer or to identify the red fact ( ). Therefore, the absence of the blue fact in C' will induce *no change* in this model's answer or its ability to select the red fact ( ). Further, to return a second supporting fact, the model would choose the next best matching fact ( ) that also indicates the start of a war and thus appears to be a reasonable choice (see bottom-right of Figure 2). Without considering interaction between the two identified facts, this model would not realize that the light blue fact ( ) does not fit well with the red fact ( ) because of the year mismatch (1950 vs. 1947), and the two together are thus insufficient to answer Q.

A good multifact model, on the other hand, connects information across different supporting facts. Thus, when evaluated on context C' with the blue fact ( ) missing, its answer as well as behavior for selecting the other supporting facts *will* be affected.

4.1 Contrastive Support Sufficiency

The above example illustrates that *sufficiency of supporting facts* in the input context matters to a good multifact model (i.e., it behaves differently under C and C') but not to a disconnected reasoning model. This suggests that if we force models

to pay attention to sufficiency, we can reduce disconnected reasoning. We formalize this idea and introduce the notion of *contrastive support sufficiency*. Informally, for each question, we consider several variants of the context that are contrastive: some contain sufficient information (i.e., $F_s \subseteq C$) while others don't. Evaluating models with these contrastive inputs allows discerning the difference in behavior between good and bad models. Figure 2 illustrates an example of contrastive contexts and the expected behavior of such models.

4.2 Transforming Existing Datasets

To operationalize this idea, we introduce an *automated* dataset transformation, Contrastive Support Sufficiency Transform ($\mathbb{T}$), applicable to any multifact reading comprehension dataset where each question is associated with a set of facts as context, of which a subset is annotated as supporting facts. Intuitively, given a context C, we want the model to identify whether C is sufficient to answer the question. If sufficient, we also want it to provide correct outputs for other tests (e.g., answer prediction).

Formally, $\mathbb{T}(D)$ transforms each instance $q = (Q, C; A)$ in a dataset D into a *group* $\mathbb{T}(q)$ of two types of instances, those with sufficient support and those without. For simplicity, consider the case of $F_s = \{f_1, f_2\}$ as in Section 3.2. The transformed instance group $\mathbb{T}(q)$ is illustrated in the bottom half of Figure 2. It includes two *insufficient context instances* corresponding to the two non-empty proper subsets of F_s, with the output label set to $L_{\text{suff}} = 0$ (illustrated as $\times$ in Figure 2):

$$(Q, C \setminus \{f_1\};\ L_{\text{suff}}{=}0), (Q, C \setminus \{f_2\};\ L_{\text{suff}}{=}0)$$

Since these contexts lack sufficient information, we omit labels for answer or supporting facts.

$\mathbb{T}(q)$ also includes a single *sufficient context instance*, but not with entire C as the context. To avoid introducing a context length bias relative to the above two instances, we remove from C a fixed, uniformly sampled non-supporting fact f_r chosen from $C \setminus F_s$ (we assume $|C| \geq 3$). The output label is set to $L_{\text{suff}} = 1$. Since the context is sufficient, the correct answer and supporting facts are included as additional labels, to use for Answer and Support tests if desired, resulting in the instance:

$$(Q, C \setminus \{f_r\};\ L_{\text{ans}}{=}A, L_{\text{supp}}{=}F_s, L_{\text{suff}}{=}1) \quad (9)$$

For any performance metric $m_\tau(q, \cdot)$ of interest in D (e.g., answer EM, support F1, etc.), the corresponding **transformed metric** $m^{\mathbb{T}}_{\tau+\text{suff}}(q, \cdot)$ operates in a conditional fashion: it equals 0 if any L_{suff} label in the group is predicted incorrectly, and equals $m_\tau(q_{\text{suff}}, \cdot)$ otherwise, where q_{suff} denotes the unique sufficient context instance in $\mathbb{T}(q)$. A model that predicts all instances to be insufficient (or sufficient) will get 0 pts under $m^{\mathbb{T}}_{\tau+\text{suff}}$.

The case of $|F_s| \geq 2$ is left to Appendix A.3. Intuitively, $m^{\mathbb{T}}_{\tau+\text{suff}}(q, \cdot) \neq 0$ suggests that when reasoning with any proper subset of F_s, the model relies on at least one supporting fact outside of that subset. High performance on $\mathbb{T}(D)$ thus suggests combining information from all facts.[7,8]

4.3 Probing Disconnected Reasoning in $\mathbb{T}(D)$

The sufficiency test (CSST) used in the transform discourages disconnected reasoning by encouraging models to track sufficiency. Much like other tests of multifact reasoning, we can apply the DIRE condition to probe models for how much they can cheat on CSST. As explained in Appendix A.4, the probe checks whether a model M can independently predict whether F_{s1} and F_{s2} are present in the input context, without relying on each other. If so, M can use disconnected reasoning to correctly predict sufficiency labels in $\mathbb{T}(D)$.

For $F_s = \{f_1, f_2\}$, if the transformed group $\mathbb{T}(q)$ uses fact f_r for context length normalization, the probing group $\mathbb{P}(\mathbb{T}(q))$ contains 3 instances:

$$(Q, C \setminus \{f_1, f_r\}; L^{?}_{\text{ans}}{=}A, L_{\text{supp}}{=}\{f_2\}, L^{*}_{\text{suff}}{=}0)$$
$$(Q, C \setminus \{f_2, f_r\}; L^{?}_{\text{ans}}{=}A, L_{\text{supp}}{=}\{f_1\}, L^{*}_{\text{suff}}{=}0)$$
$$(Q, C \setminus \{f_1, f_2\}; L^{*}_{\text{suff}}{=}-1)$$

Metric $m^{\mathbb{PT}}_{\tau+\text{suff}}(q, \cdot)$ on this probing group equals 0 if the model predicts any of the L^{*}_{suff} labels incorrectly. Otherwise, we use the grouped probe metric $m^{\mathbb{P}}_{\tau}(q, \cdot)$ from Section 3.2 for the first two instances, ignoring their L^{*}_{suff} label. Details are deferred to Appendices A.4 and A.5

We can use $\mathbb{P}(\mathbb{T}(\mathcal{D}))$ to assess how cheatable $\mathbb{T}(\mathcal{D})$ is via disconnected reasoning by computing:

$$\text{DiRe}^{\tau+\text{suff}}(\mathbb{T}(\mathcal{D})) = \text{S}^{\tau+\text{suff}}(M'^{*}, \mathbb{P}(\mathbb{T}(\mathcal{D}))) \quad (10)$$

where M'^{*} is the strongest available model architecture for τ + suff that is trained on $\mathbb{P}(\mathbb{T}(\mathcal{D}))$, and $\text{S}^{\tau+\text{suff}}$ is its score under the metric $m^{\mathbb{PT}}_{\tau+\text{suff}}$.

[7] We say *suggests* rather than guarantees because the behavior of the model with partial context $C' \subset C$ may not be qualitatively identical to its behavior with full context C.

[8] The transformation encourages a model to combine information from all facts in F_s. Whether the information that is combined is semantically meaningful or how it is combined is interesting is beyond its scope.

Dataset	Definition	Purpose	How to measure	Metric
$\mathcal{D}$	HotpotQA	Measure state of multihop reasoning	Evaluate trained M on D	$S^{\tau}(M, D)$
$\mathbb{P}(\mathcal{D})$	Probing dataset of $\mathcal{D}$	Measure how much disconnected reasoning M does on τ test of D	Evaluate M on $\mathbb{P}_{\tau}(\mathcal{D})$ in zero-shot setting or with inoculation	DiRe$^{\tau}(M, \mathcal{D})$ [Equation 7]
		Measure how much cheatable τ test of $\mathcal{D}$ is via disconnected reasoning	Train and evaluate a strong NLP model on $\mathbb{P}_{\tau}(\mathcal{D})$	DiRe$^{\tau}(\mathcal{D})$ [Equation 8]
$\mathbb{T}(\mathcal{D})$	Transformed dataset of $\mathcal{D}$	Measure truer state of multi-hop reasoning, by reducing the amount of cheatability compared to D	Evaluate trained M' on $\mathbb{T}(\mathcal{D})$	$S^{\tau+\text{suff}}(M', \mathbb{T}(\mathcal{D}))$ [Section 4.2]
$\mathbb{P}(\mathbb{T}(\mathcal{D}))$	Probing dataset of $\mathbb{T}(\mathcal{D})$	Measure how much cheatable τ+suff test of $\mathbb{T}(\mathcal{D})$ is via disconnected reasoning	Train and evaluate a strong NLP model on $\mathbb{P}_{\tau+\text{suff}}(\mathbb{T}(\mathcal{D}))$	DiRe$^{\tau+\text{suff}}(\mathbb{T}(\mathcal{D}))$ [Equation 10]

Table 1: Summary of dataset variations we create, their purposes and how we use them. τ can be any test, but our experiments are with Ans + Supp. M and M' are models that can take τ and τ + suff tests respectively. In our experiments, they are trained on D and $T(D)$ respectively with supervision for τ and τ + suff respectively.

5 Experiments

To obtain a more realistic picture of the progress in multifact reasoning, we compare the performance of the original Glove-based baseline model (Yang et al., 2018) and a state-of-the-art transformer-based LM, XLNet (Yang et al., 2019) on the multi-hop QA dataset HotpotQA (Yang et al., 2018). While it may appear that the newer models are more capable of multifact reasoning (based on answer and support prediction tasks), we show most of these gains are from better exploitation of disconnected reasoning. Our proposed transformation reduces disconnected reasoning exploitable by these models and gives a more accurate picture of the state of multifact reasoning. To support these claims, we use our proposed dataset probes, transformations, and metrics summarized in Table 1.

Datasets $\mathcal{D}$ and $\mathbb{T}(\mathcal{D})$: HotpotQA is a popular multi-hop QA dataset with about 113K questions which has spurred many models (Nishida et al., 2019; Xiao et al., 2019; Tu et al., 2020; Fang et al., 2020). We use the *distractor* setting where each question has a *set* of 10 input paragraphs, of which two were used to create the multifact question. Apart from the answer span, each question is annotated with these two supporting paragraphs and the supporting sentences within them. As described in Sec. 4.2, we use these supporting paragraph annotations as F_s to create a transformed dataset $\mathbb{T}(\mathcal{D})$.[9]

Models: We evaluate two models: **(1) XLNet-Base:** Since HotpotQA contexts are 10 paragraphs long, we use XLNet, a model that can handle contexts longer than 1024 tokens. We train XLNet-Base to predict the answer, supporting sentences, supporting paragraphs, and the sufficiency label (only on transformed datasets). As shown in Table 2 of Appendix B.4, our model is comparable to other models of similar sizes on the HotpotQA dev set. **(2) Baseline:** We re-implement the baseline model from HotpotQA. It has similar answer scores and much better support scores than the original implementation (details in Appendix B).

Metrics: We report metrics for standard tests for HotpotQA: answer span prediction (Ans), support identification (paragraph-level: Supp_p, sentence-level: Supp_s), as well as *joint* tests Ans +Supp_p and Ans +Supp_s. For each of these, we show F1 scores, but trends are similar for EM scores.[10] These metrics correspond to $m_{\tau}(q, \cdot)$ in Section 3 and to $S^{\tau}(M, D)$ in Table 1. When evaluating on the probing or transformed datasets, we use the corresponding metrics shown in Table 1.

Measuring Disconnected Reasoning

We first use our DiRe probe to estimate the amount of disconnected reasoning in **HotpotQA models** (Eqn. 7). For this, we train our models on $\mathcal{D}$ and evaluate them against $\mathbb{P}(\mathcal{D})$, the probe dataset, under three settings: (1) zero-shot evaluation (no training on $\mathbb{P}(\mathcal{D})$), (2) after fine-tuning on 1% of $\mathbb{P}(\mathcal{D})$, and (3) after fine-tuning on 5% of $\mathbb{P}(\mathcal{D})$. Since the model has never seen examples with the modified context used in the probe, the goal of fine-tuning or *inoculation* (Liu et al., 2019) is to allow the model to adapt to the new inputs, while not straying far from its original behavior on $\mathcal{D}$.

Figure 3 summarizes the results. The total heights of the bars depict overall scores of the baseline and XLNet models on $\mathcal{D}$. The upper,

[9] We do not use the sentence-level annotations as we found them to be too noisy for the purposes of transformation.

[10] See Appendix F for these metrics for all our results.

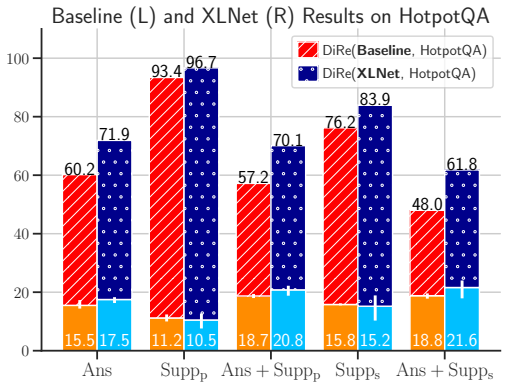

Figure 3: F1 scores for two models under various metrics. Progress on HotpotQA from Baseline model to XLNet (entire bars) is largely due to progress in disconnected reasoning (upper, darker regions), with little change in multifact reasoning (lower, ligher regions).

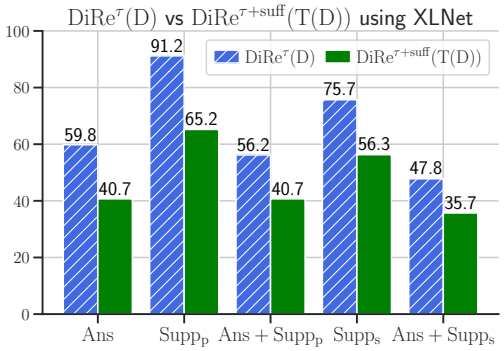

Figure 4: F1-based DiRe scores of $\mathcal{D}$ and $\mathbb{T}(\mathcal{D})$ using XLNet-Base. Dataset transformation reduces disconnected reasoning bias, demonstrated by DiRe scores being substantially lower on $\mathbb{T}(\mathcal{D})$ than on $\mathcal{D}$.

darker regions depict the portion of the overall score achieved via disconnected reasoning as estimated by the DiRe probe.[11] Their height is based on the average across the three fine-tuning settings, with white error margins depicting min/max. Importantly, results vary only marginally across the 3 settings. The lower, lighter regions show the remaining score, attributable to multifact reasoning.

First, the amount of multifact reasoning in XLNet is low—ranging from 10.5 to 21.6 F1 across the metrics. Second, even though the scores have improved going from the baseline model to XLNet, the amount of multifact reasoning (lighter regions at the bottom) has barely improved. Notably, while the XLNet model improves on the Ans + Supp$_p$ metric by 14 pts, the amount of multifact reasoning has only increased by 3 pts! While existing metrics would suggest substantial progress in multifact reasoning for HotpotQA, the DiRe probe shows that this is likely not the case—empirical gains are mostly due to higher disconnected reasoning.

As a sanity check, we also train a Single-Fact XLNet model (Appendix B.2) that only reasons over one paragraph at a time—a model incapable of multifact reasoning. This model achieves nearly identical scores on $\mathcal{D}$ as $\mathbb{P}(\mathcal{D})$, demonstrating that our DiRe probe captures the extent of disconnected reasoning performed by a model (see Appendix E).

Next, we use the DiRe probe to estimate how cheatable is the **HotpotQA dataset** via disconnected reasoning (Eqn. 8). For this, we train and evaluate the powerful XLNet model on $\mathbb{P}(\mathcal{D})$.[12]

[11] This is a *conditional* score as explained in Eqn. (7).

[12] The use of even stronger models is left to future work.

While the answer prediction test is known to be cheatable, we find that even the supporting fact (paragraph/sentence) identification test is highly cheatable (up to 91.2 and 75.7 F1, resp.).

Reducing Disconnected Reasoning

Our automatic transformation reduces disconnected reasoning bias in the dataset and gives a more realistic picture of the state of multifact reasoning. We show this by comparing how much score can a strong model (XLNet) achieve using disconnected reasoning on the original dataset, by training it on $\mathbb{P}(\mathcal{D})$ and computing DiRe$^{\tau}(D)$ (Eqn. 8), and on the transformed dataset, by training it on $\mathbb{P}(\mathbb{T}(\mathcal{D}))$ and computing DiRe$^{\tau+\text{suff}}(\mathbb{T}(\mathcal{D}))$ (Equation 10). Training the model allows it to learn the kind of disconnected reasoning needed to do well on these probes, thus providing an upper estimate of the cheatability of $\mathcal{D}$ and $\mathbb{T}(\mathcal{D})$ via disconnected reasoning.

Figure 4 shows that the XLNet model's DiRe score on the Ans +Suff metric for $\mathbb{T}(\mathcal{D})$ is only 40.7, much lower compared to its DiRe score of 59.8 on Ans for $\mathcal{D}$. Across all metrics, $\mathbb{T}(\mathcal{D})$ is significantly less exploitable via disconnected reasoning than $\mathcal{D}$, drops ranging from 12 to 26 pts.

$\mathbb{T}(\mathcal{D})$ is a Harder Test of Multifact Reasoning

By reducing the amount of exploitable disconnected reasoning in $\mathbb{T}(\mathcal{D})$, we show that our transformed dataset is harder for models that have relied on disconnected reasoning. Figure 5 shows that the transformed dataset is harder for both models across all metrics. Since a true-multihop model would naturally detect insufficient data, the drops in performance on $\mathbb{T}(\mathcal{D})$ show that the current model architectures when trained on $\mathcal{D}$ are reliant on dis-

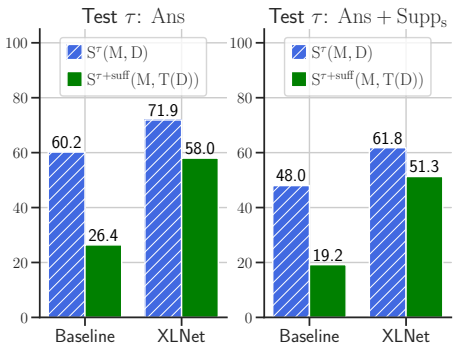

Figure 5: F1 scores of two models on $\mathcal{D}$ and $\mathbb{T}(\mathcal{D})$ under two common metrics. Transformed dataset is harder for both models since they rely on disconnected reasoning. The weaker, Baseline model drops more as it relies more heavily on disconnected reasoning.

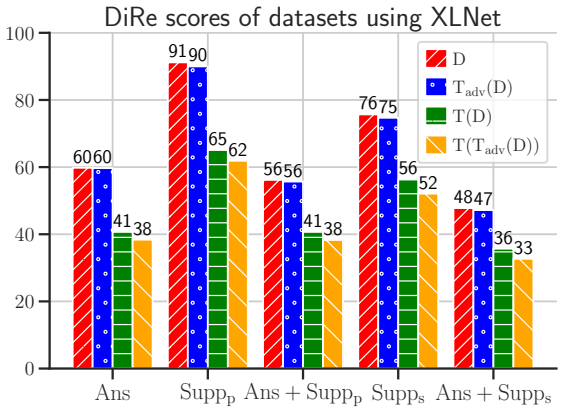

Figure 6: F1-based DiRe score on various metrics using XLNet-base for $\mathcal{D}$, adversarial $\mathbb{T}_{adv}(\mathcal{D})$, transformed $\mathbb{T}(\mathcal{D})$, and transformed adversarial $\mathbb{T}(\mathbb{T}_{adv}(\mathcal{D}))$. Transformation here is more effective than, and complementary to, adversarial augmentation.

connected reasoning. The weaker baseline model has substantially lower scores on $\mathbb{T}(\mathcal{D})$, suggesting that simple models cannot get high scores.

Single-Fact XLNet (the model incapable of multifact reasoning as described earlier) also sees a big drop (-23 F1 pts on Ans) going from $\mathcal{D}$ to $\mathbb{T}(\mathcal{D})$ – almost all of which was caught as disconnected reasoning by our DiRe probe (see Appendix E).

$\mathbb{T}(\mathcal{D})$ is Hard for the Right Reasons

Our transformation makes two key changes to the original dataset $\mathcal{D}$: (C1) adds a new *sufficiency test*, and (C2) uses a grouped metric over a set of *contrastive examples*. We argue that these changes by themselves do not result in a score drop independent of the model's ability to perform multifact reasoning (details in Appendices D and G).

Transformation vs. Adversarial Augmentation

An alternate approach to reduce disconnected reasoning is via adversarial examples for single-fact models. Jiang and Bansal (2019) proposed such an approach for HotpotQA. As shown in Figure 6, our transformation results in a larger reduction in disconnected reasoning across all metrics; e.g., the XLNet model only achieves a DiRe score (metric: Ans +Supp$_s$) of 36 F1 on $\mathbb{T}(\mathcal{D})$ as compared to 47 F1 on $\mathbb{T}_{adv}(\mathcal{D})$, computed using Eqns. (10) and (8), resp. Moreover, since our approach can be applied to any dataset with supporting fact annotations, we can even transform the adversarial dataset, further reducing the DiRe score to 33 F1.

6 Conclusions

Progress in multi-hop QA under the reading comprehension setting relies on understanding and quantifying the types of undesirable reasoning current models may perform. This work introduced a formalization of disconnected reasoning, a form of bad reasoning prevalent in multi-hop models. It showed that a large portion of current progress in multifact reasoning can be attributed to disconnected reasoning. Using a notion of contrastive sufficiency, it showed how to automatically transform existing support-annotated multi-hop datasets to create a more difficult and less cheatable dataset that results in reduced disconnected reasoning.

Our probing and transformed dataset construction assumed that the context is an unordered set of facts. Extending it to a *sequence* of facts (e.g., as in MultiRC (Khashabi et al., 2018)) requires accounting for the potential of new artifacts by, for instance, carefully replacing rather than dropping facts. Additionally, for factual reading comprehension datasets where the correct answer can be arrived at without consulting all annotated facts in the input context, our probe will unfairly penalize a model that uses implicitly known facts, even if it correctly connects information across these facts. However, our transformation alleviates this issue: a model that connects information will have an edge in determining the sufficiency of the given context. We leave further exploration to future work.

It is difficult to create large-scale multihop QA datasets that do not have unintended artifacts, and it is also difficult to design models that do not exploit such shortcuts. Our results suggest that carefully devising tests that probe for desirable aspects of multifact reasoning is an effective way forward.

Acknowledgments

This work was supported in part by the National Science Foundation under grants IIS-1815358 and CCF-1918225. Computations on beaker.org were supported in part by credits from Google Cloud. We thank the anonymous reviewers and Greg Durrett for feedback on earlier versions of this paper.

References

Akari Asai, Kazuma Hashimoto, Hannaneh Hajishirzi, Richard Socher, and Caiming Xiong. 2020. Learning to retrieve reasoning paths over Wikipedia graph for question answering. In *ICLR*.

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*.

Rajarshi Das, Ameya Godbole, Dilip Kavarthapu, Zhiyu Gong, Abhishek Singhal, Mo Yu, Xiaoxiao Guo, Tian Gao, Hamed Zamani, Manzil Zaheer, et al. 2019. Multi-step entity-centric information retrieval for multi-hop question answering. In *Proceedings of the 2nd Workshop on Machine Reading for Question Answering*, pages 113–118.

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

Yuwei Fang, Siqi Sun, Zhe Gan, Rohit Pillai, Shuohang Wang, and Jing jing Liu. 2020. Hierarchical graph network for multi-hop question answering. In *EMNLP*.

Shi Feng, Eric Wallace, Alvin Grissom II, Mohit Iyyer, Pedro Rodriguez, and Jordan Boyd-Graber. 2018. Pathologies of neural models make interpretations difficult. 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 Khashabi, Kevin Lin, Jiangming Liu, Nelson F. Liu, Phoebe Mulcaire, Qiang Ning, Sameer Singh, Noah A. Smith, Sanjay Subramanian, Reut Tsarfaty, Eric Wallace, Ally Quan Zhang, and Ben Zhou. 2020. Evaluating nlp models via contrast sets. *ArXiv*, abs/2004.02709.

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*.

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

Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel Bowman, and Noah A Smith. 2018. Annotation artifacts in natural language inference data. In *NAACL-HLT*.

Robin Jia and Percy Liang. 2017. Adversarial examples for evaluating reading comprehension systems. In *EMNLP*.

Yichen Jiang and Mohit Bansal. 2019. Avoiding reasoning shortcuts: Adversarial evaluation, training, and model development for multi-hop QA. In *ACL*.

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

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 *NAACL*.

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

Kevin Lin, Oyvind Tafjord, Peter Clark, and Matt Gardner. 2019. Reasoning over paragraph effects in situations. In *MRQA@EMNLP*.

Nelson F. Liu, Roy Schwartz, and Noah A. Smith. 2019. Inoculation by fine-tuning: A method for analyzing challenge datasets. In *NAACL-HLT*.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. 2018. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *EMNLP*.

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

Kosuke Nishida, Kyosuke Nishida, Masaaki Nagata, Atsushi Otsuka, Itsumi Saito, Hisako Asano, and Junji Tomita. 2019. Answering while summarizing: Multi-task learning for multi-hop QA with evidence extraction. In *ACL*.

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

Saku Sugawara, Pontus Stenetorp, Kentaro Inui, and Akiko Aizawa. 2020. Assessing the benchmarking capacity of machine reading comprehension datasets. In *AAAI*.

Ming Tu, Kevin Huang, Guangtao Wang, Jui-Ting Huang, Xiaodong He, and Bufang Zhou. 2020. Select, answer and explain: Interpretable multi-hop reading comprehension over multiple documents. In *AAAI*.

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.

Yunxuan Xiao, Yanru Qu, Lin Qiu, Hao Zhou, Lei Li, Weinan Zhang, and Yong Yu. 2019. Dynamically fused graph network for multi-hop reasoning. In *ACL*.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. XLNet: Generalized autoregressive pretraining for language understanding. In *NeurIPS*.

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

A Probe and Transformation Details

A.1 Probes and Transformation for $|F_s| = 2$

Figure 7 summarizes in a single place all probes and transformation discussed for the case of two supporting facts ($|F_s| = 2$).

A.2 Need for Considering All Bi-partitions

Figure 8 illustrates support bi-partitions and two examples of disconnected reasoning for a 3-hop reasoning question. It highlights the need for considering *every* bi-partition in the DIRE condition. For instance, if we only consider partitions that separate the purple ( ) and yellow ( ) facts, then the model performing the lower example of disconnected reasoning would not be able to output the correct labels in any partition and would thus appear to not satisfy the DIRE condition. We would therefore not be able to detect that it is doing disconnected reasoning.

A.3 Transformation for $|F_s| \geq 2$

The Contrastive Support Sufficiency Transform described in Section 4.2 for the case of two supporting facts can be generalized as follows for $|F_s| \geq 2$. There are two differences. First, there are $2^{|F_s|} - 2$ choices of proper subsets of F_s that can be removed to create insufficient context instances. Second, these subsets are of different sizes, potentially leading to unintended artifacts models can exploit. Hence, we use *context length normalization* to ensure every context has precisely $|C| - |F_s| + 1$ facts. To this end, let F_r be a fixed, uniformly sampled subset of $C \setminus F_s$ of size $|F_s| - 1$ that we will remove for the sufficient context instance.[13] Further, for each non-empty insufficient context $F_{s1} \subset F_s, F_{s1} \neq \phi$, let F_{r1} denote a fixed, uniformly sampled subset of F_r of size $|F_s| - |F_{s1}| - 1$. The transformed group $\mathbb{T}(q)$ contains the following $2^{|F_s|} - 1$ instances:

$$(Q, C \setminus F_r;\ L_{\text{ans}}{=}A, L_{\text{supp}}{=}F_s, L_{\text{suff}}{=}1) \quad (11)$$

$$(Q, C \setminus (F_{s1} \cup F_{r1});\ L_{\text{suff}}{=}0) \quad \text{for all } F_{s1} \quad (12)$$

Note that $|F_r| = |F_{s1}| + |F_{r1}| = |F_s| - 1$ by design, and therefore all instances have exactly $|C| - |F_s| + 1$ facts in their context.

Similar to the case of $|F_s| = 2$, for any performance metric $m_\tau(q, \cdot)$ of interest in D (e.g., answer EM, support F1, etc.), the corresponding *transformed metric* $m^{\mathbb{T}}_{\tau+\text{suff}}(q, \cdot)$ operates in a conditional fashion: it equals 0 if any L_{suff} label in the group is predicted incorrectly, and equals $m_\tau(q_{\text{suff}}, \cdot)$ otherwise, where q_{suff} denotes the unique sufficient context instance in $\mathbb{T}(q)$.

A.4 Probing $\mathbb{T}(D)$ for $|F_s| = 2$

A model M meets the DIRE condition for CSST when given an input context C', it can correctly predict whether: (i) C' contains F_{s1}, even when F_{s2} is not in C'; (ii) C' contains F_{s2}, even when F_{s1} is not in C'; and (iii) C' contains neither F_{s1} nor F_{s2}. Intuitively, if M can do this correctly, then it has the information needed to correctly identify support sufficiency for all instances in the transformed group $\mathbb{T}(q)$, without relying on interaction between F_{s1} and F_{s2}.

This leads to the following probe for $\mathbb{T}(D)$, denoted $\mathbb{P}_{\text{ans+supp+suff}}(\mathbb{T}(D))$ (sometimes simply $\mathbb{P}(\mathbb{T}(D))$ for brevity) and described here for the case of $F_s = \{f_1, f_2\}$.[14] Let f_r be the fact used for context length normalization in the transformed group $\mathbb{T}(q)$. Similar to Eqns. (1) and (2) in Section 3.2, the probing dataset contains a group $\mathbb{P}(\mathbb{T}(q))$ of instances corresponding to the unique bi-partition $\{\{f_1\}, \{f_2\}\}$ of F_s:

$$(Q, C \setminus \{f_1, f_r\}; L^{?}_{\text{ans}}{=}A, L_{\text{supp}}{=}\{f_2\}, L^{*}_{\text{suff}}{=}0)$$

$$(Q, C \setminus \{f_2, f_r\}; L^{?}_{\text{ans}}{=}A, L_{\text{supp}}{=}\{f_1\}, L^{*}_{\text{suff}}{=}0)$$

$$(Q, C \setminus \{f_1, f_2\}; L^{*}_{\text{suff}}{=}-1)$$

$L^{?}_{\text{ans}}$, as before, is an optional label that is included in the instance only if A is present in the supporting facts retained in the context of that instance.

We use the notation L^{*}_{suff} here to highlight that this label is semantically different from L_{suff} in $\mathbb{T}(D)$, in the sense that when $L^{*}_{\text{suff}} = 0$, the model during this probe is expected to produce the partial support and the answer (if present in the context). When not even partial support is there, the output label is $L^{*}_{\text{suff}} = -1$ and we don't care what the model outputs as the answer or supporting facts. Note that the label semantics being different is not an issue, as the probing method involves training models on the probe dataset.

The *joint* grouped metric here considers the sufficiency label, along with any standard test(s) of interest (answer prediction, support identification, or both). Denoted $m^{\mathbb{PT}}_{\tau+\text{suff}}(q, \cdot)$, it is defined as follows: similar to the conditional nature of the transformed metric $m^{\mathbb{T}}_{\tau+\text{suff}}(q, \cdot)$, a model receives

[13] We assume $|C| \geq 2|F_s| - 1$.

[14] Appendix A.5 describes the probe for $|F_s| \geq 2$.

Original Dataset D
$\Rightarrow$ Question $q = (Q, C; A)$ in D is assumed to be annotated with supporting facts $\{f_1, f_2\}$.

Probing Dataset $\mathbb{P}_{\text{ans+supp}}(D)$ for Answer Prediction and Support Identification tests:
$\Rightarrow$ Probing question collection $\mathbb{P}_{\text{ans+supp}}(q)$ has only one group, corresponding to the unique bi-partition $\{\{f_1\}, \{f_2\}\}$, containing:

1. $(Q, C \setminus \{f_1\}; L^{?}_{\text{ans}}{=}A, L_{\text{supp}}{=}\{f_2\})$
2. $(Q, C \setminus \{f_2\}; L^{?}_{\text{ans}}{=}A, L_{\text{supp}}{=}\{f_1\})$

Transformed Dataset $\mathbb{T}(D)$ for evaluating Constrastive Support Sufficiency:
$\Rightarrow$ Transformed question group $\mathbb{T}(q)$ in $\mathbb{T}(D)$ is defined using a single replacement fact $f_r \in C \setminus \{f_1, f_2\}$:

1. $(Q, C \setminus \{f_r\};\ L_{\text{ans}}{=}A, L_{\text{supp}}{=}F_s, L_{\text{suff}}{=}1)$
2. $(Q, C \setminus \{f_1\};\ L_{\text{suff}}{=}0)$
3. $(Q, C \setminus \{f_2\};\ L_{\text{suff}}{=}0)$

Probing Dataset $\mathbb{P}_{\text{ans+supp+suff}}(\mathbb{T}(D))$ for all three tests:
$\Rightarrow$ Probing question collection $\mathbb{P}_{\text{ans+supp+suff}}(\mathbb{T}(q))$ for the transformed question $\mathbb{T}(q)$ has only one group, corresponding to the unique bi-partition $\{\{f_1\}, \{f_2\}\}$, and is defined as:

1. $(Q, C \setminus \{f_1, f_r\};\ L^{?}_{\text{ans}}{=}A, L_{\text{supp}}{=}\{f_2\}, L^{*}_{\text{suff}}{=}0)$
2. $(Q, C \setminus \{f_2, f_r\};\ L^{?}_{\text{ans}}{=}A, L_{\text{supp}}{=}\{f_1\}, L^{*}_{\text{suff}}{=}0)$
3. $(Q, C \setminus \{f_1, f_2\};\ L^{*}_{\text{suff}}{=}-1)$

Figure 7: Proposed dataset transformation and probes for the case of $|F_s| = 2$ supporting facts.

a score of 0 on the above group if it predicts the L^{*}_{suff} label incorrectly for any instance in the group. Otherwise, we consider only the partial support instances (those with $L^{*}_{\text{suff}} = 0$) in the group, which we observe are identical to the un-transformed probe group $\mathbb{P}_{\text{ans+supp}}(q; \{f_1\})$ when ignoring the sufficiency label, and apply the grouped probe metric $m^{\mathbb{P}}_{\tau}$ from Section 3.2 to this subset of instances.

A.5 Probing $\mathbb{T}(D)$ for $|F_s| \geq 2$

The probe for disconnected reasoning in the transformed dataset $\mathbb{T}(D)$ described in Appendix A.4 for the case of two supporting facts can be generalized as follows for $|F_s| \geq 2$. For each proper bi-partition $\{F_{s1}, F_{s2}\}$ of F_s, we consider two partial contexts, $C \setminus F_{s1}$ and $C \setminus F_{s2}$, and one where not even partial support is present, $C \setminus (F_{s1} \cup F_{s2})$.

Recall that when constructing $\mathbb{T}(D)$, we had associated non-supporting facts F_{r1} and F_{r2} (both chosen from F_r) with supporting facts F_{s1} and F_{s2}, respectively, and had additionally removed them from the respective input contexts for length normalization. For the partial context instances in the probe, we choose another non-supporting fact $f_{r1} \in F_r \setminus \cup F_{r1}$, and combine it with F_{r1} to obtain $F'_{r1} = F_{r1} \cup \{f_{r1}\}$; similarly define F'_{r2}.

For each $q \in D$, the probing dataset contains a collection $\mathbb{P}(\mathbb{T}(q))$ of $2^{|F_s|-1} - 1$ groups of instances, where each group corresponds to one proper bi-partition of F_s. For the bi-partition $\{F_{s1}, F_{s2}\}$, the group, denoted $\mathbb{P}(\mathbb{T}(q); F_{s1})$, contains the following instances, each of which has exactly $|C| - |F_s|$ facts in its context:

1. $(Q, C \setminus (F_{s1} \cup F'_{r1});$
   $L^{?}_{\text{ans}}{=}A, L_{\text{supp}}{=}F_{s2}, L^{*}_{\text{suff}}{=}0)$
2. $(Q, C \setminus (F_{s2} \cup F'_{r2});$
   $L^{?}_{\text{ans}}{=}A, L_{\text{supp}}{=}F_{s1}, L^{*}_{\text{suff}}{=}0)$
3. $(Q, C \setminus F_s;\ L^{*}_{\text{suff}}{=}-1)$

The semantics of $L^{?}_{\text{ans}}$ and L^{*}_{suff} remain the same as for the case of $|F_s| = 2$.

The grouped metric for this bi-partition, denoted $m^{\mathbb{PT}}_{\tau+\text{suff}}(q, \mu(\mathbb{P}(\mathbb{T}(q); F_{s1})))$, captures whether the model exhibits correct behavior on the entire group (as discussed for the case of $|F_s| = 2$). The overall probe metric, $m^{\mathbb{PT}}_{\tau+\text{suff}}(q, \cdot)$, continues to follow Eqn. (5) and captures the disjunction of undesirable behavior across all bi-partitions.

B XLNet QA Model Details

B.1 XLNet-Base (Full)

We concatenate all 10 paragraphs together into one long context with special paragraph marker token `[PP]` at the beginning of each paragraph and special sentence marker token at the beginning of each sentence in the paragraph. Lastly, the question is concatenated at the end of this long context.

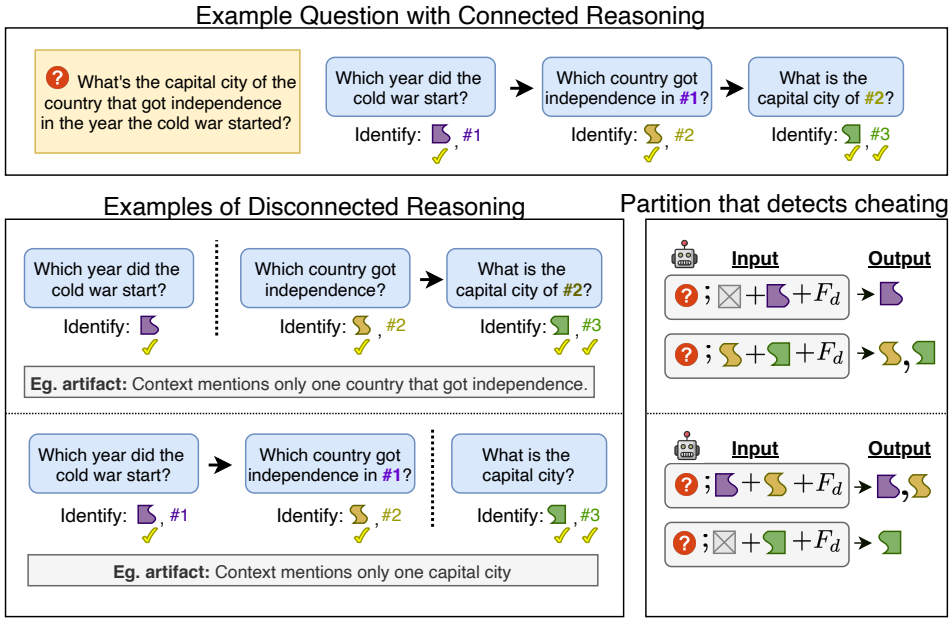

Figure 8: Generalization of disconnected reasoning to a 3-fact reasoning question. As shown in the bottom half, a model could perform multifact reasoning on two disjoint partitions to answer this question. We consider such a model to be performing disconnected reasoning as it does not use the entire chain of reasoning and relies on artifacts (specifically, it uses 1-fact and 2-fact reasoning, but not 3-fact reasoning). For each of the two examples, there exists a fact bi-partition (shown on the right) that we can use to detect such reasoning as the model would continue to produce all the expected labels even under this partition.

Apart of questions that have answer as a span in the context, HotpotQA also has comparison questions for which the answer is "yes" or "no" and it's not contained in the context. So we also prepend text "<yes> <no>" to the context to deal with both types of questions directly by answer span extraction. Concretely, we have, [CLS] <yes> <no> [PP] [SS] sent1,1 [SS] sent1,2 [PP] [SS] sent2,1 [QQ] q.

We generate logits for each paragraph and sentence by passing marker tokens through feedforward network. Supporting paragraphs and sentences are supervised with binary cross entropy loss. Answer span extraction is using standard way (Devlin et al., 2019) where span start and span end logits are generated with feedforward on each token and it's supervised with cross entropy loss. We use first answer occurrence among of the answer text among the supporting paragraphs as the correct span. This setting is very similar to recent work (Beltagy et al., 2020), and our results in Table 2, show that this model achieves comparable accuracy to other models with similar model complexity. We haven't done any hyperparameter (learning rate, num epoch) tuning on the development set because of the expensive runs, which could explain the minor difference.

For predicting sufficiency classification, we use feedforward on [CLS] token and train it with cross entropy loss. In our transformed dataset, because HotpotQA has K=2, there are twice the number of instances with insufficient supporting information than the instances with insufficient supporting information. So during *training* we balance the number of insufficient instances by dropping half of them.

B.2 XLNet-Base (Single Fact)

To verify the validity of our tests, we also evaluate a variant of XLNet *incapable of Multifact reasoning*. Specifically, we train our XLNet model that makes predictions one paragraph at a time (similar to Min et al. (2019)). Although these previous works showed that answer prediction is hackable, we adapt it to predict supporting facts and sufficiency as well.

Specifically, we process the following through the XLNet transformer [CLS] <yes> <no> [PP] [SS] sent1,1 [SS] sent1,2 [QQ] q for each paragraph. We then supervise [PP] tokens for two tasks: identify if paragraph is a supporting paragraph and identify if paragraph has the answer span (for yes/no question both supporting paragraphs are supervised to be having

the answer). We then select top ranked paragraph for having the answer and generate the best answer span. Similarly, select top two ranked paragraphs for having being supporting and predict the corresponding supporting sentences. The logits for answer span and supporting sentences are ignored when the paragraph doesn't have the answer and is not supporting respectively. We train for three losses jointly: (i) ranking answer containing paragraph, (ii) ranking supporting paragraphs (iii) predicting answer from answer containing paragraph (iv) predicting supporting sentences from supporting paragraphs. We use binary cross entropy for ranking of paragraphs, so there's absolutely no interaction the paragraphs in this model. To get the sufficiency label, we apply check if the sufficiency classification label based on the number of supporting paragraphs predicted[15]. For original dataset, if $|\text{predicted}(\text{Supp}_\text{p})| > 1$, then $C = 1$ otherwise $C = 0$. For probing dataset, if $|\text{predicted}(\text{Supp}_\text{p})| > 0$, then $C = 0$ otherwise $C = -1$.

B.3 Glove-based Baseline

We have re-implemented the baseline described in (Yang et al., 2018) in AllenNLP (Gardner et al., 2017) library. Unlike original implementation, which uses only answer and sentence support identification supervision, we also using paragraph supervision identification supervision. Additionally, we use explicit paragraph and sentence marker tokens as in our XLNet-based implementation, and supervise model to predict paragraph and sentences support logits via feedforward on these token marker representations. We train answer span identification by cross-entropy loss and both paragraph and sentence support identification with binary cross-entropy loss.

B.4 QA Model Results

Table 2 shows results for QA models. Our XLNet model is comparable to other models of similar sizes on the HotpotQA dev set. Our implementation of RNN baseline model has answer scores similar to the reported ones, and has much better support identification scores than the original implementation.

[15]This heuristic exploits the fixed number of hops=2 and doesn't need any training on the sufficiency label. We use this heuristic because we want to predict sufficiency label without interaction across *any* of the facts.

Model	Ans F1	Supp_s F1	Joint F1
Baseline (reported)	58.3	66.7	40.9
QFE (BERT-Base)	68.7	84.7	60.6
DFGN (BERT-Base)	69.3	82.2	59.9
RoBERTa-Base	73.5	83.4	63.5
LongFormer-Base	74.3	84.4	64.4
Baseline (our)	60.2	76.2	48.0
XLNet-Base	71.9	83.9	61.8

Table 2: Performance of XLNet-Base compared to other transformer models (of similar size) on HotpotQA. Our model scores higher than BERT-Base models QFE (Nishida et al., 2019) and DFGN (Xiao et al., 2019), and performs comparable to recent models using RoBERTa and Longformer (Beltagy et al., 2020).

C Implementation and Model Training

All our models are implemented using AllenNLP (Gardner et al., 2017) library. For XLNet-base, we have also used Huggingface Transformers (Wolf et al., 2019). For all XLNet-base experiments, we train for two epochs, checkpointing every 15K instances and early stopping after 3 checkpoints of no validation metric improvement. For Glove-based baseline model, we do the same but for 3 epochs. For both models, effective batch size were 32. For XLNet-based model, we used learning rate of 0.00005 and linear decay without any warmup. The hyper-parameters were chosen as the default parameters used by hugging-face transformers to reproduce BERT results on SQuAD dataset. Our experiments were done using V100 gpus from Google Cloud. On average XLNet training runs took 2 days on 1 gpu and baseline model took less than 1 day.

D Human Evaluation of Sufficiency Prediction

The sufficiency test can cause a spurious drop if sufficiency labels are incorrect, i.e., the context is sufficient even after f_1 or f_2 is removed. To rule this out, we randomly evaluated (using MTurk) 115 paragraphs from $C \setminus F_s$, and found only 2 (1.7%) could be used in place of f_1 or f_2 to answer the question. As we show below, this would result in only a marginal score drop compared to the roughly 20% observed drops.

To estimate this, we setup an annotation task on MTurk for turkers to annotate whether a pair of facts has sufficient information to arrive at an answer. For each question, we create three pair (f_1, f_2), (f_1, f_r) and (f_r, f_2), where f_1 and f_2 are annotated supporting paragraphs, and f_r is a

Question: Did Greg Costikyan have the same profession as John Dolmayan? **Answer**: no		
f1: John Hovig Dolmayan (Armenian: Ջոն Հովսեփյան , born July 15, 1973) is a Lebanese-born Armenian–American songwriter and drummer. He is best known as the drummer of System of a Down. Dolmayan is also the drummer for the band Indicator and former drummer for Scars on Broadway. His energetic live performances with System Of A Down over the years, have garnered him critical acclaim. Loudwire listed him as one of the "Top 50 Hard Rock + Metal Drummers Of All Time" , with Dolmayan being ranked at #22.	**f2**: Greg Costikyan (born July 22, 1959, in New York City), sometimes known under the pseudonym "Designer X", is an American game designer and science fiction writer.	**fr**: System of a Down, sometimes shortened to System and abbreviated as SOAD, is an Armenian-American heavy metal band from Glendale, California, formed in 1994. The band currently consists of Serj Tankian (lead vocals, keyboards), Daron Malakian (vocals, guitar), Shavo Odadjian (bass, backing vocals) and John Dolmayan (drums).
Question: Both Dusty Drake and Joe Diffie sing which genre of music? **Answer:** country		
f1: Dean Buffalini (born February 23, 1965) is an American country music artist, known professionally as Dusty Drake. Drake played various venues in his native Pennsylvania for several years before moving to Nashville, Tennessee, co-writing a 1996 single for Joe Diffie. By 2003, Drake was signed to Warner Bros. Records as a recording artist. That year, he released three singles from his self-titled debut album, including "One Last Time", his first Top 40 entry on the Hot Country Songs charts. Drake released a fourth single for the label before exiting in 2004.	**f2**: Joe Logan Diffie (born December 28, 1958) is an American country music singer. After working as a demo singer in the 1980s, he signed with Epic Records\' Nashville division in 1990. Between then and 2004, Diffie charted 35 cuts on the "Billboard" Hot Country Songs chart, including five number one singles: his debut release "Home", "If the Devil Danced (In Empty Pockets)", "Third Rock from the Sun", "Pickup Man" (his longest-lasting number one, at four weeks) and "Bigger Than the Beatles". In addition to these cuts, he has 12 other top ten singles and ten other top 40 hits on the same chart. He also co-wrote singles for Holly Dunn, Tim McGraw, and Jo Dee Messina, and has recorded with Mary Chapin Carpenter, George Jones, and Marty Stuart.	**fr**: My Give a Damn\'s Busted is a song written by American country music artist Joe Diffie along with Tom Shapiro and Tony Martin. Diffie originally recorded the song on his 2001 album "In Another World". The song was later recorded by Jo Dee Messina on her album "Delicious Surprise". Released on January 3, 2005, Messina\'s version spent two weeks at the top of the "Billboard" Hot Country Songs charts that year, and her first chart single since "I Wish" in late 2003 – early 2004. Canadian country music singer Michelle Wright included her version of the song on her 2006 album "Everything and More".

Figure 9: Two examples where we found a non-supporting fact provides an alternative support for answering the question. **f1** and **f2** are annotated supporting facts, but **fr** in C and **f1** form alternative support.

randomly sample from total non-supporting paragraphs. The questions were taken from HotpotQA development set. If for a question, annotators agree that both (f_1, f_2) and (f_1, f_r) are sufficient, we assume f_r provides proxy (duplicate) information for f_2. Likewise for (f_1, f_2) and (f_r, f_2).

Out of 115 examples questions (with annotator agreement), we found only 2 (1.7%) of them to have a proxy fact in f_r. Figure 9 shows these 2 examples. This shows that such proxy information is very rare in HotpotQA. We next estimate the impact of these duplicates on the human score.

D.1 Human Score Estimate on $\mathbb{T}(\mathcal{D})$

Given the number of observed duplicates, we can now estimate the expected drop in human performance. For simplicity, lets consider only the Exact Match score where the human would get one point if they predict all the facts exactly. There are two scenarios where the sufficiency test in our transformed dataset would introduce more noise resulting in a drop in human score.

1. The original context was not actually sufficient: In this case the sufficiency label, $L_{\text{suff}} = 1$ would be incorrect and the human score on the sufficiency test for this example would be zero. However, in such a case, the human score on paragraph and sentence identification would also be zero. As a result, there would be no drop in human score relative to the original task

2. The constrastive examples are actually sufficient: Due to a potential proxies of f_1 in $C \setminus f_1$, it is possible that our contrastive examples would be considered sufficient. While this would also effect the original dataset, its impact would be more extreme on our test. We focus on this scenario in more detail next.

Lets assume that there are k such proxy paragraphs in any given context. In such a case, there is a $1/(k+1)$ chance that a human would select the annotated support paragraphs instead of these proxy paragraphs. So there is a $1/(k+1)$ chance

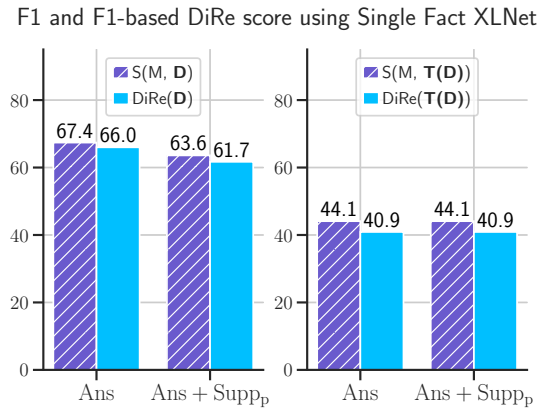

Figure 10: F1 and F1-based DiRe scores of $\mathcal{D}$ and $\mathbb{T}(\mathcal{D})$ using Single-Fact XLNet-base.

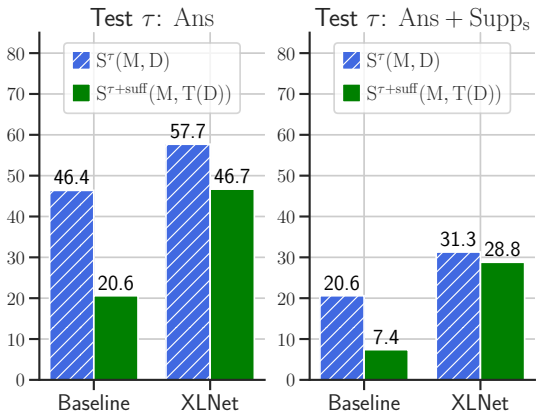

Figure 11: EM scores of two models on $\mathcal{D}$ and $\mathbb{T}(\mathcal{D})$ under two common metrics. Transformed dataset is harder for both models since they rely on disconnected reasoning. The weaker, Baseline model drops more as it relies more heavily on disconnected reasoning.

that they get one point on the original task, but they would always get 0 points on our transformation.

Given that we observed a proxy paragraph in 1.7% of our annotated paragraphs, we can model the likelihood of observing k proxy paragraphs with a binomial distribution. Specifically, since there are 8 distractor paragraphs in HotpotQA, the probability of observing k proxy paragraphs:

$$P(k) = \binom{8}{k} \times (0.017)^k \times (1 - 0.017)^{8-k}$$

So the expected drop in score would be given by:

$$\sum_{k=1}^{8} P(k) \times \frac{1}{k+1} = 0.0628$$

So the expected drop in human score is only 6.28% whereas we observed about 18% drop in EM scores as shown in Appendix F.

E XLNet (Single-Fact) Results

Figure 10 shows the results of our Single-Fact model on the original dataset $\mathcal{D}$ and the transformation $\mathbb{T}(\mathcal{D})$. On both the metrics, we can see that our DiRe probe gets the almost the same score as the Single-Fact model, i.e., our probe can detect the disconnected reasoning bias in the Single-Fact model. Additionally, we can see that score of this Single-Fact model drops from 67.4 to 44.1, a drop of 23 F1 pts, going from $\mathcal{D}$ to $\mathbb{T}(\mathcal{D})$ (on the Ans metric). This shows that our transformed dataset is less exploitable by a disconnected reasoning model.

F Exact Match Numbers

Figure 11 shows the EM scores of our models on the original dataset $\mathcal{D}$ and transformed dataset $\mathbb{T}(\mathcal{D})$. Consistent with our F1 metric, we can see large drops in model score going from $\mathcal{D}$ to $\mathbb{T}(\mathcal{D})$, showing that the transformation is harder for these models

Figure 12 shows the disconnected reasoning bias in the XLNet-Base model trained on $\mathcal{D}$ and $\mathbb{T}(\mathcal{D})$ using the EM scores. Again, we see the same trend here – the transformed dataset has a reduced DiRe score indicating lower disconnected reasoning bias.

Finally, Figure 13 shows the impact of adversarial examples and the transformation on the EM scores. While the drops are lower due to the strictness of the EM scores, the trends are still the same – adversarial examples have a minor impact on the DiRe scores but transformation of the original dataset as well transformation of the adversarial examples results in a big drop in the disconnected reasoning bias.

G Grouped Metric on Trivial Transformation

The grouped metric combines decisions over a set of instances and, one can argue, is therefore inherently harder. However, one can show that unless the instances within a group test for qualitatively different information, the grouped metric will not be necessarily lower than the single instance metric.

To support this claim, we compute grouped metric over a trivial transform that is similar to $\mathbb{T}(\mathcal{D})$ but does not involve the contrastive sufficiency prediction test. This trivial transform, denoted $\mathbb{T}_{trv}$, creates 3 copies of each instance but drops at ran-

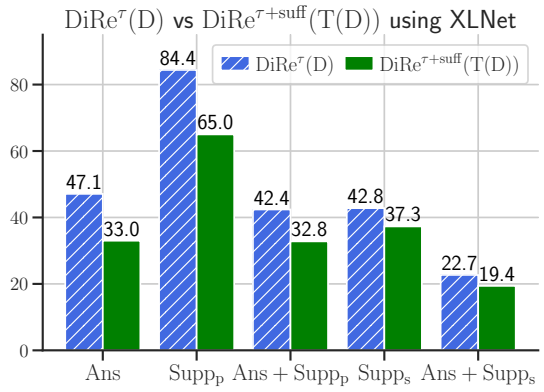

Figure 12: EM-based DiRe scores of $\mathcal{D}$ and $\mathbb{T}(\mathcal{D})$ using XLNet-Base. Dataset transformation reduces disconnected reasoning bias, demonstrated by DiRe scores being substantially lower on $\mathbb{T}(\mathcal{D})$ than on $\mathcal{D}$.

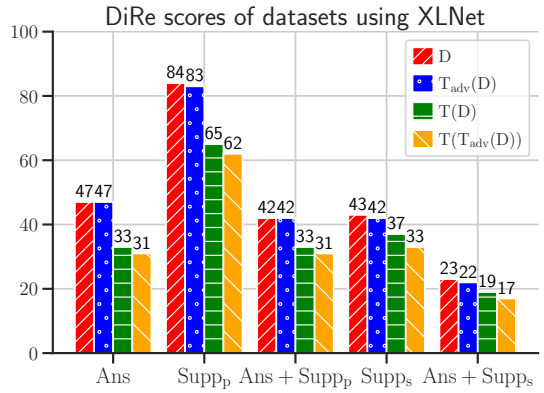

Figure 13: EM-based DiRe score on various metrics using XLNet-base for four datasets: original $\mathcal{D}$, adversarial $\mathbb{T}_{\text{adv}}(\mathcal{D})$, transformed $\mathbb{T}(\mathcal{D})$, and transformed adversarial $\mathbb{T}(\mathbb{T}_{\text{adv}}(\mathcal{D}))$. Transformation is more effective than, and complementary to, Adversarial Augmentation for reducing DiRe scores.

dom one non-supporting fact from each instance. Similar to $\mathbb{T}(\mathcal{D})$ in which we require the model to produce correct sufficiency labels for all 3 instances, here we require the model to produce correct answer and support on all 3 copies.[16]

1. $(Q, C \setminus \{f_{r_1}\};\ L_{\text{ans}}{=}A, L_{\text{supp}}{=}F_s)$
2. $(Q, C \setminus \{f_{r_2}\};\ L_{\text{ans}}{=}A, L_{\text{supp}}{=}F_s)$
3. $(Q, C \setminus \{f_{r_3}\};\ L_{\text{ans}}{=}A, L_{\text{supp}}{=}F_s)$

In Figure 14, we show the EM results corresponding to the respective grouped metrics for $\mathbb{T}_{trv}(\mathcal{D})$ and $\mathbb{T}(\mathcal{D})$). We see barely any drop of results from $\mathcal{D}$ to $\mathbb{T}_{trv}(\mathcal{D})$, but do see significant drop going from $\mathcal{D}$ to $\mathbb{T}(\mathcal{D})$. This shows that

[16]Note that our transformed dataset does not even require the answer and support labels on all the examples, making $\mathbb{T}(\mathcal{D})$, in some ways, easier than this dataset.

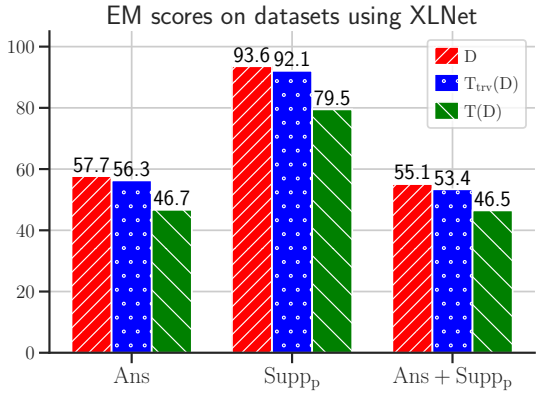

Figure 14: EM scores of XLNet-base on three metrics for original HotpotQA ($\mathcal{D}$), trivially transformed HotpotQA ($\mathbb{T}_{trv}(\mathcal{D})$) and our transformed HotpotQA ($\mathbb{T}(\mathcal{D})$). Model scores barely drop from $\mathcal{D}$ to $\mathbb{T}_{trv}(\mathcal{D})$ but significantly drop $\mathcal{D}$ to $\mathbb{T}(\mathcal{D})$ showing that drop of scores in $\mathbb{T}(\mathcal{D})$ is not simply a result of using a grouped metric

adding a grouped metric over an arbitrary set of decisions would not make $\mathbb{T}(\mathcal{D})$ harder. Hence, the drop in scores from $\mathcal{D}$ to $\mathbb{T}(\mathcal{D})$ is not simply a result of using a grouped metric.