Explaining Unverifiable Presuppositions

Neha Srikanth, Rupak Sarkar, Navita Goyal, Connor Baumler, Rushil Dandamudi
Department of Computer Science
University of Maryland, College Park
{nehasrik,rupak,navita,baumler,rushilcd}@umd.edu

Abstract

Current QA systems deal with unanswerable questions due to presupposition failure in a primitive manner, merely labeling them as "Unanswerable" yet providing little insight into why. Kim et al. (2021) attempt to bridge such a gap by building a pipeline consisting of presupposition generation and verification, followed by explanation generation utilizing the negated presupposition. While an essential step towards tackling unanswerability, their explanation generation module fails to incorporate appropriate evidence from the supporting document that justifies the failure of the presupposition. In this work, we explore several different approaches to constructing natural language explanations that leverage recent advances in claim verification, dense retrieval, and rationale generation. We conduct a human evaluation study of generated explanations and find that evidence retrieval and composition remains a challenging task for state-of-the-art retrieval models.

1 Introduction

While the vast majority of current QA models focus on correctly answering questions, many popular QA datasets such as Natural Questions (Kwiatkowski et al., 2019) also include questions that are *unanswerable*. Such questions may be unanswerable due to malformation, subjectivity, underspecified language, or their commonsensical nature (Kim et al., 2021).

This work focuses on a class of questions that are unanswerable due to *presuppositional failure*, or questions based on implicit false assumptions. Figure 1 shows an example. The question *"How many years was Símon Bolívar married to his second wife?"* is based on the false presupposition that Símon Bolívar, an early Latin American political leader, was married more than once. When left uncorrected, presupposition failures of this kind can be misleading to information-seeking users. Kim et al. (2021) present a novel pipeline to verify and produce simple explanations for false question presuppositions. However, as shown in Figure 1, their system produces explanations limited to the negation of the unverifiable presupposition. For information-seeking users, it is often necessary to go beyond pointing out presupposition failure. Instead, it is desirable to directly provide evidence pieces across multiple documents that support unanswerability, as in the evidence-based explanation at the bottom of Figure 1.

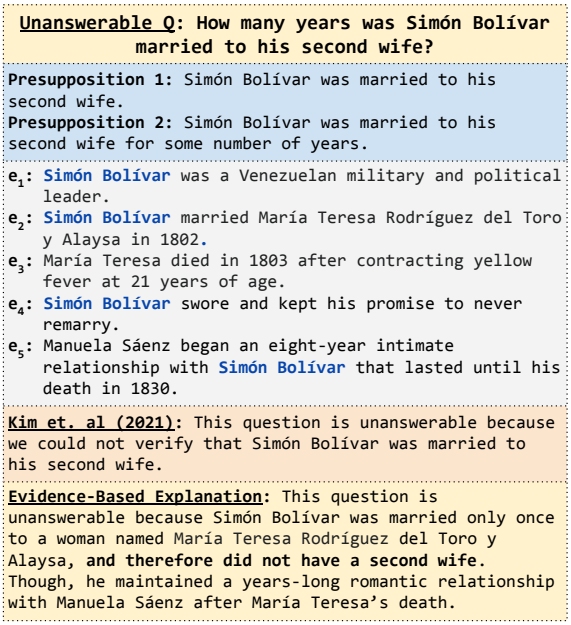

Figure 1: An example of a question that is unanswerable due to presuppositional failure. We aim to generate evidence-based explanations for false presuppositions to aid information-seeking users.

In this work, we aim to construct evidence-based explanations for unanswerable questions due to presuppositional failure that go beyond simply negating their presuppositions. Concretely, given a presupposition P generated from a question Q, we seek to generate an explanation E consisting of one or more pieces of evidence $\{e_1, e_2, ..., e_n\}$ that accurately support the presupposition in the case

of accommodation, or its negation, in the case of presuppositional failure.

In contrast to Kim et al. (2021), we do not rely directly on natural language inference (NLI) (Giampiccolo et al., 2007) models for presupposition verification (and in turn, explanation generation). Moreover, we do not limit the scope of documents considered for the verifiability of the presupposition to the gold document tagged with the question. Instead, our pipeline mainly consists of passing some query (in the form of the original unanswerable question, a related question, or a presupposition) to a document retrieval module (Karpukhin et al., 2020) to first collect a set of candidate documents or passages. Then, we directly utilize an evidence retrieval model trained on the FEVER dataset (Thorne et al., 2018) to select n evidence pieces e_i from our candidate set of documents in an extractive manner. We then construct explanations of two forms: a simple concatenation of all n evidence pieces, as well as an explanation generated by an NLI rationale generation system (Wiegreffe et al., 2021). We conclude by presenting a comparative human evaluation study of generated explanations and analyze some of the challenges of explanation generation for presuppositions.

2 Background

Here we describe the linguistic phenomena of presuppositions and their extraction through the identification of presupposition triggers.

What are presuppositions? Natural language can be an incredibly compact form of communication. In dialogue, information is sometimes introduced *indirectly* through statements or questions. Presuppositions are implicit assumptions in a question or statement taken for granted by discourse participants. For example, the utterance *"Manuela Sáenz's relationship with Simón Bolívar lasted longer than María Teresa's"* presupposes *"Simón Bolívar had a relationship with María Teresa."*

Beyond statements, questions may have presuppositions as well. The question *"Why didn't Maria Teresa's relationship with Simón Bolívar last longer than Manuela Sáenz's?"* again presupposes that *"Simón Bolívar had a relationship with María Teresa."* This particular question highlights another important property of presuppositions—they *project* out of negative environments. See Simons et al. (2010) for a more complete discussion on presupposition projection.

Presupposition Triggers. Traditionally, presuppositions are identified via lexical and syntactic *triggers* in statements, such as the word *both* in the proposition *"Both my dogs like peanut butter"*, which presupposes the existence of exactly two contextually salient dogs. Presupposition triggers have been studied extensively in linguistics literature, and the list of common presupposition triggers proposed by Levinson et al. (1983) is most widely used in practice today.

Kim et al. (2021) restrict their study of presuppositions in questions to those triggered by the six most common triggers present in *wh*-questions from the NQ dataset: question words (e.g *what*), definite articles (*the*), factive verbs (e.g. *know*), possessives (e.g. *my dog's bone*), temporal adjuncts (e.g. *after*), and counterfactuals (e.g. *if I had eaten, I wouldn't have been hungry*). Each trigger is associated with one or more templates that they then use to construct presuppositions that the authors manually validate. We broaden the scope of generated presuppositions beyond those presented in Kim et al. (2021) by expanding the trigger set to incorporate 8 more syntactic constructions from Levinson et al. (1983), studied in depth in the NOPE corpus (Parrish et al., 2021). These triggers include:

- **Change of state verbs**: Questions containing change of state verbs such as *disappear* presuppose that the subject was previously in a different state.
- **Clefts**: Cleft statements (e.g. *Was it her who ate the pudding?*) take the canonical form *"It was X that did Y"*. Such statements presuppose that *someone* or *something* did Y.
- **Comparatives**: Comparative questions usually compare two entities along a particular attribute. *"Does she have a smaller house than us?"* presupposes that *"She has a house."*
- **Aspectual Verbs:** Aspectual verbs (i.e *initiate* or *resume*) in questions indicate whether or not the event in question had or had not been happening previously. The question *"Why did she initiate an argument?"* presupposes that *"An argument had not been previously happening."*
- **Embedded Questions**: Constructions such as *"How did she know why he chose to divorce*

	Trigger Type	Example	Presupposition
Kim et al. (2021)	Question words	Why did European empires colonize Southeast Asia?	European empires colonized Southeast Asia.
	Definite article	Did the curse get broken in Season 7 of Once Upon a Time?	the curse exists in Season 7 of Once Upon a Time, the curse is contextually unique in Season 7 of Once Upon a Time
	Factive verbs	Why did she know that he didn't want her on the team?	He didn't want her on the team.
	Possessives	How did the Polar Express' conductor know whose house to stop at?	The Polar Express had a conductor.
	Temporal adjuncts	After Lisa Kudrow starred in The Comeback, did she star in the Friends Reunion?	Lisa Kudrow starred in The Comeback.
	Counterfactuals	If World War II hadn't happened, would the US's relations with Germany be the same?	World War II happened.
Parrish et al. (2021)	Change of state verbs	Why did she appear on the president's cabinet in September?	She was not a part of the president's cabinet before.
	Clefts	Was it Manuela Saenz he got married to?	He got married to someone.
	Comparatives	Which pyramid is a larger structure than the Pyramid of Giza?	The Pyramid of Giza is a large structure.
	Aspectual verbs	Why did they start promoting the movie after it was released?	They had not previously been promoting the movie.
	Embedded questions	How did the Jen Psaki not understand why the Biden administration pushed back?	The Biden Administration pushed back.
	Clause-embedding predicates	Why did Disney regret attempting to revive Lizzie McGuire?	Disney attempted to revive Lizzie McGuire.
	Implicative predicates	Why did Disney attempt to revive Lizzie McGuire?	Disney was unsuccessful at reviving Lizzie McGuire.
	Numeric determiners	Did both seasons of The Comeback star Lisa Kudrow?	There were exactly two seasons of The Comeback.
	Re-prefixed verbs	Has the United States re-entered diplomatic talks with Russia?	The United States had previously entered diplomatic talks with Russia.

Table 1: Example presupposition triggers

his wife?" embed questions into statements or other questions. The example presupposes *"He chose to divorce his wife."*

- **Clause-Embedding Predicates**: Certain clause-embedding verbs may presuppose the existence of their embedded clause. The question *"Why didn't he regret lying to his teacher?"* presupposes *"He lied to his teacher."*

- **Implicative Predicates**: Implicative predicates such as *dared* presuppose an attribute of the action in their embedded clause. For example, the question *"Why did she dare to challenge the President?"* presupposes that challenging the President is a difficult, risky, or courageous task.

- **Numeric Determiners**: Questions containing particular numeric determiners presuppose the existence of exactly some number of entities or actions. The question *"Do all three of my children need to come to the doctor?"* presuppose the existence of exactly three contextually salient children.

- **Re-verbs**: Questions containing particular *re* verbs presuppose that the described action has taken place previously. For example, the question *"Why did she retell that story?"* presupposes that *"She had told this story previously."*

We choose to augment the trigger set from Kim et al. (2021) because the trigger classes from the NOPE corpus extend beyond hard triggers, such as *"both"* which requires satisfaction of the presupposition for the question to be well-formed, to include contextually-sensitive soft presupposition triggers, such as the change of state verb *"melted"*, which can be cancelled under larger contexts. Table 1 contains more examples for each trigger type that we study from both Kim et al. (2021) and Parrish et al. (2021).

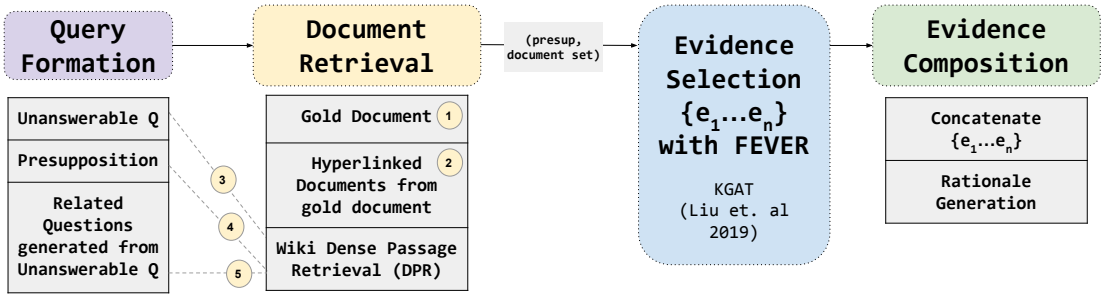

Figure 2: Our overall explanation generation pipeline.

3 Methodology

Our study focuses on the subset of questions marked as *Unanswerable* in the Natural Questions dataset and specifically explores explainability of questions that are unanswerable due to presuppositional failure.

Kim et al. (2021) frame presupposition verification as a fact verification task and use NLI models (Dagan et al., 2005) trained on the FEVER dataset (Thorne et al., 2018), treating Wikipedia documents as premises and presuppositions as hypotheses. As documents are prohibitively large inputs for NLI systems, the authors split the Wikipedia document associated with the question into n sentences to generate n premise-hypothesis pairs. They then aggregate predictions across all *(presupposition, sentence)* pairs and use a simple majority vote to determine the factuality of the presupposition. Upon determining a presupposition's factuality, they negate it, and return that as an explanation.

In contrast, we do not rely on NLI for verification and explanation, and instead use a claim verification model over documents to retrieve evidence pieces for verifiability and explanation construction. Our explanation generation pipeline consists of three modules: (1) query formation (§3.1), (2) candidate document retrieval (§3.2) (3) evidence selection (§3.3), and (4) evidence composition (§3.4). Figure 2 illustrates our overall pipeline. We experiment with different query formation and document retrieval configurations before ultimately feeding in a (query, document set) to a FEVER evidence selection module.

3.1 Query Formation

We experiment with three different query formations that are ultimately used to retrieve a candidate document set, prior to evidence selection. We hypothesize that varying the query format may impact the quality of retrieved documents. Ultimately, these queries are solely used to retrieve a set of documents. We always pass a presupposition and a candidate document set to the FEVER evidence selection module to verify.

Unanswerable Question. We pass the original unanswerable question to a Wikipedia dense passage retriever (DPR) (Karpukhin et al., 2020) fine-tuned on the NQ dataset to retrieve a set of candidate passages. While the question itself *is* known to be unanswerable, we explore whether relevant passages are still retrieved.

Presupposition Generation. Presuppositions of questions themselves may differ in language and content from the original unanswerable question. We pass presuppositions of the original answerable question to DPR to retrieve a presupposition-informed candidate set of passages.

We templatically generate an exhaustive list of presuppositions for each question by using the trigger list discussed in Section 2. For each trigger, we implement detection using `spaCy` (Honnibal et al., 2020) based on the trigger's syntactic properties. When a particular trigger is present in a question, we apply a template to generate a corresponding presupposition. Each generated presupposition is post-hoc manually edited by an expert annotator to ensure that it is both well-formed and is projected correctly. Questions may result in multiple presuppositions if they contain multiple triggers.

Related Question Generation. The query formats above make the simplifying assumption that retrieved documents must be relevant to all entities in the original unanswerable question or its presupposition. However, this could constrain the search to a small set of passages identified as relevant to the full question or presupposition, ignoring stronger evidence from other more distantly-related passages or sentences. We explore related question generation as an evidence-gathering tool, deriving

questions related to the original unanswerable question to run through the DPR system to retrieve a larger set of candidate passages. For example, our proposed question generation-based module would generate the following related questions for the unanswerable question: *"Which linguist invented the lightbulb?"*:

(1) Who invented the lightbulb?
(2) What are linguists?
(3) What was the profession of the inventor of the lightbulb?

Presented together, the answers to the questions above could constitute a reasonable explanation for the presupposition failure of the original question. We consider both template-based and neural methods to generate questions. For template-based generation, we apply a number of templates to each query primarily based on the interrogative word used. For example, starting from questions of the form *"Who X'ed Y?"*, we generate follow-up questions of the form *"Was Y X'ed?"*, as the action in question must have been done in order for it to have been done by someone. Concretely, given the question *"Who invented the lightbulb?"*, this system would generate the follow-up "Was the lightbulb invented?" For questions that start with the *wh*-word *"what"* and a form of *"is"* (e.g., "What is the difference between intramural and interscholastic sports?"), we generate a followup to ask whether the entity or concept in question exists in the first place (e.g., *"Is there a difference between intramural and interscholastic sports?"*).

These templates are designed specifically for *wh*-questions, the focus of our study, and require the use of an interrogative word to help identify the structure and meaning of the original question.

For our neural question generation model, we use RoleQGeneration (Pyatkin et al., 2021), a competitive semantic role-based QG model. RoleQGeneration is answer-agnostic, requiring as input a passage and a predicate, and generates questions associated with different semantic roles. This formulation fits nicely with the distribution and nature of questions from NQ. We treat unanswerable questions as passages, and their corresponding root verb as the predicate in question. This structure generates a finite amount of interpretable questions that can be used to clarify parts of the original unanswerable question and widen the scope of retrieved documents.

3.2 Document Retrieval

The query formats presented above are used as input to a DPR model meant to retrieve relevant Wikipedia documents to focus on for claim verification. As alternatives to DPR, we also consider constraining the document set to only the gold document as well as the set of documents hyperlinked from the gold document.

Gold Document. Each question in the NQ dataset is paired with a "gold document", meant to serve as the document containing the answer. We explore constraining the document set passed to FEVER to only the document associated with the original unanswerable question.

Hyperlinked Documents. For unanswerable questions, the gold document may not always contain evidence supporting the presupposition's falsehood. To provide additional background information for verifying presuppositions, we include Wikipedia documents hyperlinked to the main section of the gold Wikipedia document. The candidate document set comprises of 3–49 documents with an average of $\sim$ 17 documents per question.

Wikipedia Dense Passage Retrieval. Explanations for unanswerable questions' presupposition failure can not always be sourced from the paired gold document (or in some cases, hyperlinked documents). For example, the last piece of evidence e_5 in Figure 1 was not present in the Wikipedia document `Simón_Bolívar`, but rather the document `Manuela_Sáenz`. We turn to neural passage retrieval as a mechanism to collect relevant *passages* to constrain the sources that a FEVER claim verification model uses. We leverage DPR (Karpukhin et al., 2020), a retrieval system based on generating and searching dense representations of Wikipedia passages. We use the DPR model implementation from `pyserini` (Lin et al., 2021), utilizing a DPR model finetuned on the Natural Questions dataset.

3.3 Evidence Selection

For presupposition verification and explanation evidence retrieval, we use models trained on the FEVER dataset (Thorne et al., 2018). We use KGAT (Liu et al., 2020), a graph attention network used by Kim et al. (2021) as a competitive presupposition verification model.

FEVER systems typically comprise of three modules: document retrieval, sentence selection,

and natural language inference (NLI). For document retrieval, we use the methods in Section 3.2 to construct a set of documents, in the case of the gold document or hyperlinked documents, and passages in the case of DPR.

We feed generated question presuppositions as "claims" and run sentence selection models trained on the FEVER dataset (Liu et al., 2020) on the retrieved Wikipedia documents. We then select the top five relevant sentences for each presupposition to serve as pieces of evidence $\{e_1...e_5\}$. When presented together, these pieces of evidence act as an explanation for presupposition failure, as illustrated in Figure 1. Similarly, for presupposition verification, we use the KGAT NLI model, which uses a graph attention network over the retrieved evidence graph to calculate importance of each piece of evidence to verify the claim.

3.4 Evidence Composition

The evidence selection approach detailed in Section 3.3 returns a set of n sentences (in our setup, $n = 5$) and a label {`support`, `refute`, `not enough info`}. For unverifiable presuppositions, we explore two different evidence composition methods: simple concatenation and rationale generation.

Concatenation. Pure evidence-based explanations consisting of selected evidence pieces concatenated together are simple yet informative. We take the top evidence sentences returned from our FEVER pipeline and concatenate them as an explanation.

Rationale Generation. The concatenation approach discussed above is purely *extractive*. However, many question presuppositions (or their negations) may suffer from reporting bias (Gordon and Van Durme, 2013), potentially limiting the effectiveness of extractive approaches. Leveraging the conclusion from Shwartz and Choi (2020) that pretrained language models overcome reporting bias to some extent, we explore the ability of inference rationale generation models to generate coherent, relevant explanations.

We use a self-rationalizing T5 model from Wiegreffe et al. (2021) jointly predicts an entailment label, as well as produces a free-text rationale explaining the relation. The rationale generation model is finetuned on E-SNLI (Camburu et al., 2018), a rationale dataset built on top of SNLI (Bowman et al., 2015)) and COS-E, an extension of the CommonsenseQA dataset (Talmor et al., 2018) containing rationales for correct answer choices. We supply a presupposition and corresponding evidence pieces $\{e_1...e_5\}$ to the rationale generation model to understand whether a generation-based approach yields stronger explanations. While most previous work in rationale generation have primarily focused on extractive rationale generation due to their purported faithfulness (Wiegreffe et al., 2021), we use free-text models mainly for their usefulness from the point-of-view of an information-seeking user.

4 Experiments

We conduct five experiments to generate explanations for unverifiable presuppositions using the overall pipeline discussed in Section 3 with the varying query formats and document retrieval methods. We obtain the sample of 100 unanswerable *wh*-questions from the NQ development set annotated by the authors of Kim et al. (2021) in Section 3 of their paper which we refer to as `google-presup`.

4.1 Experimental Setup

We run each of the 100 unanswerable questions in `google-presup` through our augmented trigger detection and templatic presupposition generation pipeline, resulting in a total of 193 presuppositions. For each of the 193 presuppositions, an expert annotator (1) verified that the presupposition was well-formed and projected, and (2) labeled its veracity using all documents across Wikipedia. This resulted in a clean test set of 85 unverifiable presuppositions stemming from 50 unique unanswerable questions.

We run all of our experiments on the sample of 85 unverifiable presuppositions. Figure 2 contains experiment numbers in yellow circles corresponding to following five settings:

> **Experiment 1: Gold Document Retrieval.** In this baseline experiment, we constrain the candidate document set to only the gold document paired with the original unanswerable question. We pass each presupposition and the full gold document (all passages from the Wikipedia document) to the FEVER claim verification system to produce the evidence pieces concatenated together to form the explanation.
>
> **Experiment 2: Hyperlinked Document Re-**

trieval. We carefully loosen the candidate set of documents to pass to the FEVER system by collecting all hyperlinked documents from the gold document.

Experiment 3: DPR with Unanswerable Questions. We pass each original unanswerable question through a DPR system to retrieve the top 10 relevant passages across Wikipedia to feed to our FEVER system.

Experiment 4: DPR with Presuppositions. We pass each presupposition through a DPR system to retrieve the top 10 relevant passages across Wikipedia to feed to our FEVER system.

Experiment 5: DPR with Related Question Generation. For each original unanswerable question, we generate n related questions using the RoleQG question generation system. We then pass each related question through the DPR system to obtain a total of $10 * n$ passages and pool all of them together to feed to our FEVER system.

4.2 Human Evaluation

We ask humans to judge the explanations consisting of selected evidence pieces concatenated together across multiple dimensions:

- **Completeness**: When presented as a set, do the pieces of evidence form a complete explanation of the presupposition failure? This is a binary label, with 0 indicating the explanation does not explicate the presupposition failure and 1 indicating it successfully does.
- **Relevance to Presupposition**: How many pieces of evidence (out of 5) are directly relevant to the presupposition (or "claim" in the FEVER claim verification paradigm)? For example, when all five pieces of selected are relevant to entities or events mentioned in the claim, human judges may assign a value of 5.
- **Relevance to Explanations**: Just because evidence pieces are relevant to the presupposition does not mean they constitute a satisfactory explanation of its veracity. This score denotes the degree to which evidence pieces contribute to understanding whether the presupposition can be refuted or not. A value of 5 indicates that the explanations are sufficient for a human to understand why the presupposition is false.

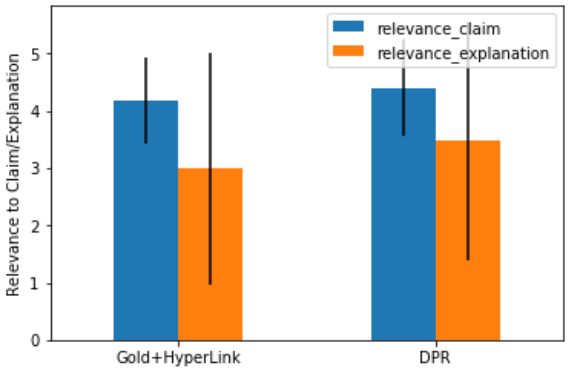

Figure 3: Relevance to Claim and Explanation

5 Results and Discussion

We select the best non-DPR experiment (Experiment 2) and best DPR-based experiment (Experiment 3) for our human evaluation study described in Section 4.2. We found the gold document setup from Experiment 1 too restricting with respect to evidence scope. The presupposition query format from Experiment 4 and semantic-role based question generation from Experiment 5 resulted in too high a volume of noisy documents (i.e for the question *How many years was Simón Bolívar married to his second wife?*, the system produced questions like *Who was married?*, *Who married?*).

5.1 Human Evaluation

In our human evaluation, we find that only 23% of the explanations are *complete* when using gold document along with its hyperlinked document set (Experiment 2). In contrast, explanations from Experiment 3 (DPR-based) yields a completeness score of 63%.

Further, we compare the relevance to the presupposition and explanation in the two settings (Figure 3). We find that the retrieved explanations are almost equally relevant in both the experiments. However, the relevance to explanation is much lower for both the experiments.

5.2 Qualitative Analysis

In some cases, our pipeline *is* able to retrieve relevant, strong evidence pieces that fit nicely together to form an explanation.

Question: Why did European Countries give up their colonies in Southeast Asia?

Presupposition: there is some reason that european countries gave up their colonies in Southeast Asia.
$\{e_1\}:$ The independence of the Thirteen colonies in North America in 1783 after the American war of Independence caused Britain to lose some of its most oldest and most populous colonies. $\{e_2\}:$ It comprised territories and colonies of the Spanish monarch in the Americas , Asia , Oceania and Africa , as the Greater Antilles ... as well as a number of Pacific Ocean archipelagos including the Philippines ; and it lasted until the early 19th century Spanish American wars of independence , which left only Cuba , Puerto Rico , and the Philippines and various territories in Africa still under Spanish rule .

While the evidence pieces do not directly negate the presupposition in the above example, the explanation does list the American War of Independence as a reason Britain "lost" its colonies, which supports the idea that they did not simply "give them up".

However, our pipelined approach suffers from a few different issues. Firstly, we find that some generated presuppositions are too commonsensical to retrieve relevant evidence for. For example, the generated presupposition *a life sentence lasts for some years in delaware* proved difficult for our system to verify. Some of the presuppositions stemming from definite articles also tended to be highly commonsensical (i.e *something is an acceptable three letter abbreviation for phenylthiocarbamide*).

In addition, much of the task of presupposition explanation revolved around verifying that a particular entity or attribute did not exist. Verifying the non-existence of entities proved to be an difficult task for our claim verification system, and it instead retrieved evidence in support of an entity's existence.

Question Generation Analysis. We use the RoleQGeneration model to generate related questions from unanswerable questions to aid in evidence-gathering. The original model bases questions on sentences' action verbs. Unanswerable questions with these types of verbs resulted in coherent and useful related questions. However, questions with copular verbs needed extra care (prototypes) for question generation; such generated questions were not as useful.

For sentences with predicates not trained in the RoleQGeneration model, a user may input their own *prototype* question to output for some input sentences. These prototypes are template questions with the filler token `<something>`. The model then relies on generating such a question by replacing `<something>` with the appropriate relevant words from the input text. For example, with a prototype template of "`What/who is something?`" and an input sentence *"The dog is fast"*, the system would generate "What / who is fast?"

We provide some example of the generated questions in Table 2. The model generates 2–4 related questions from the original input.

The results above indicate the model's dependence on the main predicate. While the first example exemplifies useful related questions (based on the predicate "invent") that could gather strong evidence when fed into an end-to-end QA system, the next few questions that rely on the predicates ("is" or "has") either perpetuate the false presupposition or are trivial. Questions 2–4 illustrate the model's sensitivity to the input form. For example, question 2 is a simple restatement of the question, while questions 3 and 4 resulted in relevant related questions.

6 Related Work

Presupposition. Presuppositions have been studied extensively in linguistics literature. Perhaps the most widely accepted theory on presupposition, Strawson (1950) proposed presuppositions as preconditions for the truth or falsity of utterances. Since then, there have been attempts to enumerate presupposition triggers (Levinson et al., 1983; Potts, 2015) as well as properties of presuppositions such as projection (Van der Sandt, 1992; Beaver and Krahmer, 2001), cancellation (Simons et al., 2010), and gradience (Tonhauser et al., 2018). Computationally, Cianflone et al. (2018) were the first to study detection of presupposition trigger contexts at scale. Kim et al. (2021) study presuppositions within the context of QA, and Parrish et al. (2021) construct a large scale dataset of presuppositions from sentence-context pairs to study how well NLI models capture presuppositional relationships. Presuppositions are present in a large portion of NQ questions, and serve as a useful framework rooted in deep linguistic theory for explaining question unanswerability.

Explanation and Rationale Generation. Explainability of NLP systems has become an increasingly studied area as competitive pre-trained models dominate a large number of tasks (Wiegreffe and Marasovic, 2021). Several natural language explanation-based datasets and models have been introduced for NLI-related tasks; Camburu et al. (2018) introduce a dataset of natural lan-

Question	RoleQ Generated Questions
Which linguist invented the lightbulb?	Who invented the lightbulb? What did linguist invent?
What is the tenth album of the 21 Pilots?	What is the tenth album of the 21 Pilots?
Does the 21 Pilots have a tenth album?	Who has a tenth album? What do the 21 Pilots have?
No one knows if the 21 Pilots have a tenth album.	Who has a tenth album? Does the 21 Pilots have a tenth album? Does no one have a tenth album? Who knows if the 21 Pilots has as tenth album?

Table 2: RoleQG examples

guage explanations of entailment relations for the SNLI (Bowman et al., 2015) dataset, and Brahman et al. (2020) introduce a crowdsourced dataset of explanations for defeasible NLI. Kotonya and Toni (2020) study explanability within domain-specific fact checking, and within QA, Rajani et al. (2019) develop explanation generation models for commonsense question answering to explain answer choice selection. In a similar manner, we add a component of explainability to the existing task of presupposition verification.

7 Conclusion

In this paper, we explore whether false presuppositions occurring in unanswerable Natural Questions (Kwiatkowski et al., 2019) can be explained in a way that can help an information-seeking human understand why the question is unanswerable. Through several approaches detailed in Section 4, the broad conclusion we draw is that existing systems fail to explain false presuppositions in a satisfiable manner.

References

David Beaver and Emiel Krahmer. 2001. A partial account of presupposition projection. *Journal of logic, language and information*, 10(2):147–182.

Samuel R Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. 2015. A large annotated corpus for learning natural language inference. *arXiv preprint arXiv:1508.05326*.

Faeze Brahman, Vered Shwartz, Rachel Rudinger, and Yejin Choi. 2020. Learning to rationalize for nonmonotonic reasoning with distant supervision. *arXiv preprint arXiv:2012.08012*.

Oana-Maria Camburu, Tim Rocktäschel, Thomas Lukasiewicz, and Phil Blunsom. 2018. e-snli: Natural language inference with natural language explanations. *Advances in Neural Information Processing Systems*, 31.

Andre Cianflone, Yulan Feng, Jad Kabbara, and Jackie Chi Kit Cheung. 2018. Let's do it" again": A first computational approach to detecting adverbial presupposition triggers. *arXiv preprint arXiv:1806.04262*.

Ido Dagan, Oren Glickman, and Bernardo Magnini. 2005. The pascal recognising textual entailment challenge. In *Machine Learning Challenges Workshop*, pages 177–190. Springer.

Danilo Giampiccolo, Bernardo Magnini, Ido Dagan, and Bill Dolan. 2007. The third PASCAL recognizing textual entailment challenge. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, pages 1–9, Prague. Association for Computational Linguistics.

Jonathan Gordon and Benjamin Van Durme. 2013. Reporting bias and knowledge acquisition. In *Proceedings of the 2013 workshop on Automated knowledge base construction*, pages 25–30.

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. spaCy: Industrial-strength Natural Language Processing in Python.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online. Association for Computational Linguistics.

Najoung Kim, Ellie Pavlick, Burcu Karagol Ayan, and Deepak Ramachandran. 2021. Which linguist invented the lightbulb? presupposition verification for question-answering. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3932–3945, Online. Association for Computational Linguistics.

Neema Kotonya and Francesca Toni. 2020. Explainable automated fact-checking for public health claims. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7740–7754, Online. Association for Computational Linguistics.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466.

Stephen C Levinson, Stephen C Levinson, and S Levinson. 1983. *Pragmatics*. Cambridge university press.

Jimmy Lin, Xueguang Ma, Sheng-Chieh Lin, Jheng-Hong Yang, Ronak Pradeep, and Rodrigo Nogueira. 2021. Pyserini: A Python toolkit for reproducible information retrieval research with sparse and dense representations. In *Proceedings of the 44th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2021)*, pages 2356–2362.

Zhenghao Liu, Chenyan Xiong, Maosong Sun, and Zhiyuan Liu. 2020. Fine-grained fact verification with kernel graph attention network. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7342–7351, Online. Association for Computational Linguistics.

Alicia Parrish, Sebastian Schuster, Alex Warstadt, Omar Agha, Soo-Hwan Lee, Zhuoye Zhao, Samuel R. Bowman, and Tal Linzen. 2021. NOPE: A corpus of naturally-occurring presuppositions in English. In *Proceedings of the 25th Conference on Computational Natural Language Learning*, pages 349–366, Online. Association for Computational Linguistics.

Christopher Potts. 2015. Presupposition and implicature. *The handbook of contemporary semantic theory*, 2:168–202.

Valentina Pyatkin, Paul Roit, Julian Michael, Yoav Goldberg, Reut Tsarfaty, and Ido Dagan. 2021. Asking it all: Generating contextualized questions for any semantic role. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1429–1441, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Nazneen Fatema Rajani, Bryan McCann, Caiming Xiong, and Richard Socher. 2019. Explain yourself! leveraging language models for commonsense reasoning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4932–4942, Florence, Italy. Association for Computational Linguistics.

Vered Shwartz and Yejin Choi. 2020. Do neural language models overcome reporting bias? In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6863–6870, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Mandy Simons, Judith Tonhauser, David Beaver, and Craige Roberts. 2010. What projects and why. In *Semantics and linguistic theory*, volume 20, pages 309–327.

Peter F Strawson. 1950. On referring. *Mind*, 59(235):320–344.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2018. Commonsenseqa: A question answering challenge targeting commonsense knowledge. *arXiv preprint arXiv:1811.00937*.

James Thorne, Andreas Vlachos, Oana Cocarascu, Christos Christodoulopoulos, and Arpit Mittal. 2018. The Fact Extraction and VERification (FEVER) shared task. In *Proceedings of the First Workshop on Fact Extraction and VERification (FEVER)*.

Judith Tonhauser, David I Beaver, and Judith Degen. 2018. How projective is projective content? gradience in projectivity and at-issueness. *Journal of Semantics*, 35(3):495–542.

Rob A Van der Sandt. 1992. Presupposition projection as anaphora resolution. *Journal of semantics*, 9(4):333–377.

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

Sarah Wiegreffe, Ana Marasović, and Noah A. Smith. 2021. Measuring association between labels and free-text rationales. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10266–10284, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.