*Proceedings of the 13th Conference on Language Resources and Evaluation (LREC 2022)*, pages 4552–4560
Marseille, 20-25 June 2022

# A STEP towards Interpretable Multi-Hop Reasoning: Bridge Phrase Identification and Query Expansion

**Fan Luo, Mihai Surdeanu**
University of Arizona
Tucson, AZ, USA
fanluo@email.arizona.edu, msurdeanu@email.arizona.edu

## Abstract

We propose an unsupervised method for the identification of bridge phrases in multi-hop question answering (QA). Our method constructs a graph of noun phrases from the question and the available context, and applies the Steiner tree algorithm to identify the minimal sub-graph that connects all question phrases. Nodes in the sub-graph that bridge loosely-connected or disjoint subsets of question phrases due to low-strength semantic relations are extracted as bridge phrases. The identified bridge phrases are then used to expand the query based on the initial question, helping in increasing the relevance of evidence that has little lexical overlap or semantic relation with the question. Through an evaluation on HotpotQA(Yang et al., 2018), a popular dataset for multi-hop QA, we show that our method yields: (a) improved evidence retrieval, (b) improved QA performance when using the retrieved sentences; and (c) effective and faithful explanations when answers are provided.

## 1. Introduction

Multi-hop question answering (QA) requires synthesizing information across multiple documents or paragraphs to infer the correct answer. Information retrieval (IR) techniques are commonly used to narrow down the search space from millions of web pages to a small set of relevant documents or paragraphs. However, simply matching the semantics of the question and context is not sufficient to answer a multi-hop question. Considering the example in Figure 1. For the question "*What is the name of the executive producer of the film that has a score composed by Jerry Goldsmith?*", it is necessary to first find the film that *'has a score composed by Jerry Goldsmith'*. And then find out the *'name of the executive producer'* of the film. Phrases such as *'Alien'* that connect these supporting evidences in multi-hop QA are called **bridge phrases**.

Identifying bridge phrases remains one of the challenging problems for multi-hop QA, especially when explainability is a concern. In this work, we propose an unsupervised method for bridge phrase identification, and show that it improves all downstream components in a multi-hop QA system. In particular, the contributions of this paper are:

**(1)** We introduce a new strategy, for the identification of bridge phrases for multi-hop QA by dynamically connecting scattered information pieces from the question and the available context, organizing them as a noun phrases graph, modeling the bridge phrases identification task as a Steiner tree problem (Hartmanis, 1982) [1], and then applying Takahashi and others (1980)'s algorithm to identify the minimal sub-graph that connects all question phrases, in which the Steiner points are extracted as bridge phrases. We call our method **STEP** (**S**teiner **T**ree-based **E**xpansion **P**hrase identification).

**Question:** What is the name of the executive producer of the film that has a score composed by Jerry Goldsmith?

**Supporting Document1: Alien (soundtrack)**
The iconic, avant-garde score to the film "Alien" was composed by Jerry Goldsmith and is considered by some to be one of his best, most visceral scores. . .

**Supporting Document2: Alien (film)**
Alien is a 1979 science-fiction horror film ... Dan O'Bannon, drawing upon previous works of science fiction and horror, wrote the screenplay from a story he co-authored with Ronald Shusett ... Shusett was executive producer.
. . .

**Bridge Phrases:** Alien

**Answer:** Ronald Shusett

Figure 1: An example multi-hop question from HotpotQA (Yang et al., 2018). Information pieces from two supporting documents need to be connected in order to infer the answer. However, there is no lexical overlap between the question and the sentence containing the answer, which can be used by a general-purpose retriever to locate all the supporting information pieces, especially the answer *'Ronald Shusett'*. To answer this question, an interpretable question answering approach needs first identify the bridge phrase *'Alien'*, which is *'the film that has a score composed by Jerry Goldsmith',* and then answer the simpler question *"What is the name of the executive producer of the film Alien?"*

**(2)** Our method is modular and agnostic to any downstream multi-hop QA components. We show how the output of our algorithm can be used to expand the query used for evidence retrieval, as well as to provide enhanced context for the answer extraction component. Lastly, we show how our method can be used to provide post-hoc explanations to provided answers.

[1] `https://en.wikipedia.org/wiki/Steiner_tree_problem`

**(3)** We evaluate our method on the HotpotQA dataset (Yang et al., 2018), and show that: (a) it improves evidence retrieval for both traditional information retrieval methods such as BM25 (Trotman et al., 2014) and neural retrievers (Reimers and Gurevych, 2019), (b) it yields better answer extraction performance, and (c) it generates better explanation for provided answers.

## 2. Related Work

Recent research on multi-hop QA has proposed multiple different strategies. We group them in the following three categories:

**(1) Question decomposition:** Min et al. (2019) proposed DecompRC, a system that learns to break multi-hop questions into simpler, single-hop sub-questions. Jiang and Bansal (2019) proposed a controller RNN which decomposes the multi-hop question into multiple single-hop sub-questions, and dynamically inferred a series of reasoning modules.

**(2) Question reformulation coupled with iterative evidence retrieval:** Feldman and El-Yaniv (2019) proposed a method to iteratively retrieve supporting paragraphs by forming a joint vector representation of both questions and paragraphs. In each subsequent retrieval iteration, they use the paragraphs retrieved in the previous iteration to reformulate the search vector. Asai et al. (2020) used a joint encoding of the question and current passage to iteratively retrieve a subsequent passage in the reasoning chain with an RNN. Qi et al. (2019) introduced GoldEn Retriever, which is trained to generate a query from the question and the available context at each reasoning step. Das et al. (2019) proposed a multi-step reasoner, which reformulates the question into its latent space with respect to its current value and the state of the reader.

**(3) Query Expansion Techniques for QA:** Most query expansion (QE) approaches expand the original query with terms obtained based on (a) a thesaurus, including synonyms, hypernyms, and acronyms (Esposito et al., 2020; Nakade et al., 2018), (b) an ontology, a knowledge base describing the concepts, properties, instances, and structure of knowledge of a domain (Guo et al., 2018; Wang et al., 2017; Alromima et al., 2016), or (c) additional features, such as query logs, web search information, or user intent mined from a community QA archive (Azad and Deepak, 2019; Bouadjenek et al., 2016; Wu et al., 2014).

**(4) Graph-based multi-hop QA:** Recent studies built entity graphs from multiple paragraphs, and applied graph neural networks (GNNs) to conduct reasoning over these graphs (Cao et al., 2019; Xiao et al., 2019). Tu et al. (2019) introduced a Heterogeneous Document-Entity graph, including nodes corresponding to candidates, documents and entities. Fang et al. (2019) created a hierarchical graph with nodes on different levels of granularity (questions, paragraphs, sentences, entities), the representations of which are initialized with pre-trained contextual encoders. CogQA Ding et al. (2019) iteratively extracted entities and answer candidate spans for each hop and organized them as a cognitive graph.

Our work stands on the intersection of the latter two categories. However, unlike most Graph-based multi-hop QA methods proposed in the above-cited papers, our method is unsupervised. Besides, instead of training a complex end-to-end black-box model , our method focuses on identifying the bridge phrases used by query expansion to boost the performance of evidence retrieval. Importantly, even though our method operates over a noun-phrase graph to identify bridge phrases, it does not require or expect the answer to be a noun phrase as many other graph-based methods do.

## 3. Approach

In this section, we provide an overview of STEP, and show how it fits into existing state-of-the-art retriever-reader QA frameworks (Chen et al., 2017). Figure 2 overviews our approach; Table 1 shows a walk-through example.

### 3.1. Noun Phrase Extraction

We extract noun phrases from the input question and $N$ candidate paragraphs (documents)[2]. To extract the noun phrases from the input question, we apply: (1) quotation extraction, i.e., extracting noun phrases between a pair of single or double quotes; (2) noun phrase grounding, using the titles from all the paragraphs as a simple encyclopedic resource and fuzzy match to tolerate typo and variations; (3) basic normalization, i.e., removing special punctuation and wh-words like *'what' and 'who'*; (4) named entity recognition, such as *'New York'* with a named entity type of *GPE*; and (5) noun chunks extraction, i.e., extracting chunk of text containing a single noun word (*'songwriter'*) or a noun plus the words describing the noun (*'an organic compound'*). All these noun phrases are normalized by: converting them to lower case, removing articles, lemmatization, and ignoring transparent question words such as *'time'*, *'place'*, *'event'*, etc.

We use the same process to extract noun phrases from every paragraph, with the exception that we apply coreference resolution before all the steps listed above, and use both document titles and extracted question phrases as the encyclopedias for noun phrase grounding. In addition, we add the prefix *[$P_i$]*[3] to 'polyseme' named entities[4] (such as '*[$P_8$] January*'), and

[2] Paragraphs or documents depends on the input dataset. For HotpotQA, we use paragraph(s) and document(s) interchangeably, because HotpotQA only extracts the introductory paragraphs of documents from Wikipedia.

[3] The prefix *[$P_i$]* indicates the phrase is extracted from the $i_{th}$ paragraph.

[4] We consider a named entity with an entity type that is one of 'DATE', 'LANGUAGE', 'NORP', 'GPE', 'CARDINAL', 'PERCENT', 'LOC', 'QUANTITY','ORDINAL' as a 'polyseme' named entity.

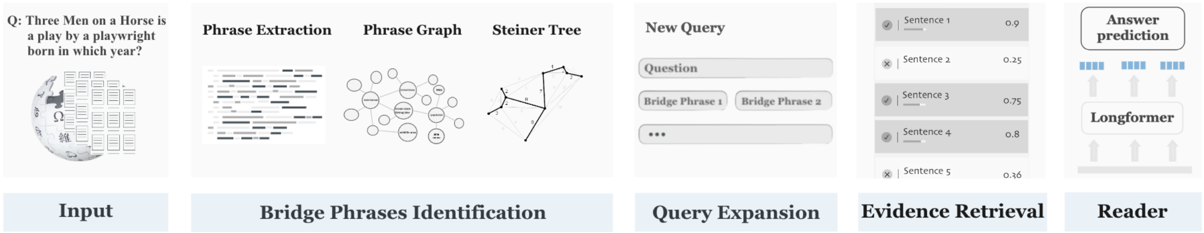


Figure 2: Overview of our approach. Given the input question and a corpus of candidate documents, we first extract noun phrases and construct the noun phrase graph. We then use the Steiner tree algorithm to identify the minimal sub-graph that connects all question phrases. The Steiner points in the graph that bridge subsets of question phrase nodes are identified as the bridge phrases. These phrases are then appended to the original question to generate an augmented query. An off-shelf retrieval model uses the augmented query to rank the context sentences, and the top ones are then used as input to a Longformer reader for answer extraction.

to noun chunks containing no named entity (such as '*[$P_5$]* video game') to distinguish the noun phrases that appear the same in text but might refer to different things. For example, '*[$P_1$] January*' extracted from paragraph 1 is *2011 January*, while '*[$P_3$] January*' from paragraph 3 refers to *2017 January*. For all the components mentioned above, we used either SpaCy (Honnibal et al., 2020) or in-house components.

## 3.2. Noun-phrase Graph Construction

After all noun phrase mentions in the paragraphs have been identified, we create the initial noun phrase graph $G$: unique noun phrase are nodes in this graph; edges encode the co-occurrence or coreference relation between noun phrases. We model three types of co-occurrence: (a) SENT-SENT edges, which capture co-occurrences between noun phrases mentioned in the same sentence (as *'George Francis Abbott' - '[$P_2$] playwriter'* from supporting document 2 in Table 1); (b) TITLE-SENT edges, which connect noun phrases occurring in the title of the document and its most similar noun phrase from each sentence or any single word noun chunk in this document; and (c) TITLE-TITLE edges, which connect noun phrases extracted from the same title. For example, *'Tomb Raider' - '2013 video game'* when the title is *'Tomb Raider (2013 video game)'*. To better capture coreference, even though we already applied coreference resolution when extracting noun phrases, we also add: (d) coreference edges between inclusive phrases from the same paragraph (as *'Ronald Shusett' - 'Shusett'* in Figure 1).

Note that the initial noun phrase graph $G$ can be disjoint, since some paragraphs do not share noun phrases with others. We then prune $G$ by discarding disconnected graph components that do not contain a node that matches with any of the question phrases. If a question phrase is not in $G$, but a node in G fuzzily matches or partially matches the question noun phrase or a node matches with the question noun phrase when removing the prefix *[$P_i$]*, we add the question phrase as a new node and add an edge between the new question phrase node and the node matches with it. For example, we add edge 'Blake Shelton song' - 'Blake Shelton' for the question phrase 'Blake Shelton', and add edge '*[$P_1$] play' - 'play'* for the question phrase 'play' as in Table 1.

Instead of using entity linking to identify which phrases are similar to question phrases, which requires an external knowledge base (KB) for the mapping, we run fuzzy matching to find out the nodes that represent context noun phrases that are similar to the question phrases. If there are more than one disjoint graph components containing at least one node that match with a question phrase, we then add additional edge between nodes that are same without the prefix, such as '*[$P_2$] voice' - '[$P_5$] voice'*. We call the resulting graph the Relevant Graph ($RG$).

## 3.3. Steiner Tree Computation

We frame the identification of bridge phrases as a *minimum Steiner tree* problem, i.e., we use the Steiner algorithm (Takahashi and others, 1980) to compute the minimum spanning tree of the sub-graph that contains all question phrases. Our implementation runs NetworkX's approximation minimum Steiner tree algorithm[5] over $RG$ to identify the Steiner points (i.e., bridge phrases) that do not exist in the question but are needed to connect all question phrases in the graph. For example, our algorithm identifies *'George Abbott'* and *'George Francis Abbott'* as the bridge phrases for the question shown in Table 1.

## 3.4. Query Expansion and Retrieval

In multi-hop QA, the subsequent evidence beyond the first hop often fails to be retrieved, due to little lexical overlap or semantic relation with the question. To bridge the information gap inherent between multi-hop questions and their answers, we adopt the query expansion technique to augment the question with the bridge phrase(s) proposed by STEP. As an example, for the question in Table 1, we expand the question to *"Three Men on a Horse is a play by a playwright born in which year, George Abbott, George Francis Abbott"*, so that

[5] `https://networkx.org/documentation/stable/_modules/networkx/algorithms/approximation/steinertree.html`

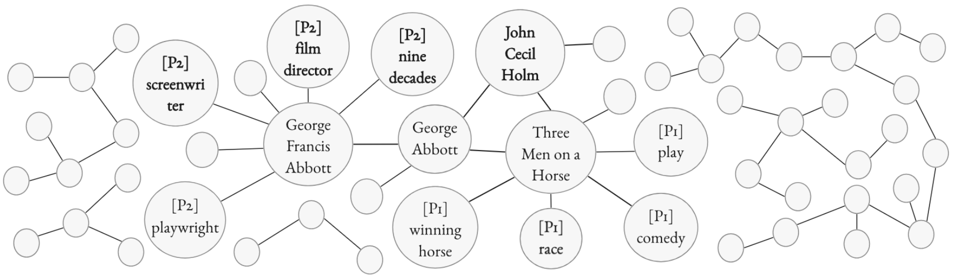
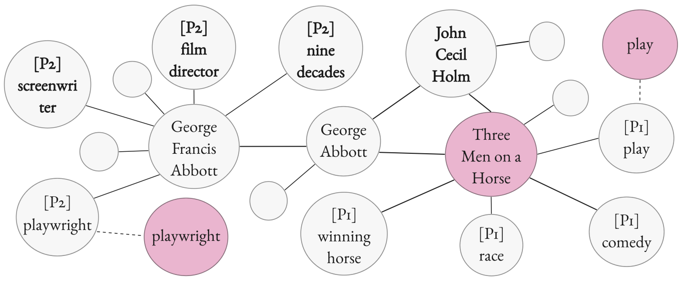
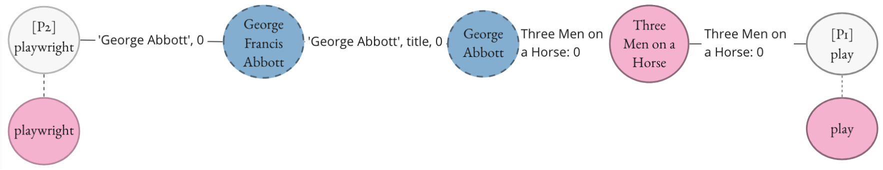
| | |
|---|---|
| Input | **Question**: Three Men on a Horse is a *play* by a *playwright* born in which year?<br>**Supporting Document 1: Three Men on a Horse**<br>Three Men on a Horse is a *play* by George Abbott and John Cecil Holm. . . .<br>**Supporting Document 2: George Abbott**<br>George Francis Abbott (June 25, 1887 – January 31, 1995) was an American theater producer and director, *playwright*, screenwriter, and film director and producer whose career spanned nine decades.<br>. . . |
| STEP$_1$:<br>Noun phrase Extraction | **Question**: [Three Men on a Horse]$_G$ is a [play]$_C$ by a [playwright]$_C$ born in which year?<br>***Supporting Document 1:*** [***Three Men on a Horse***]$_T$<br>[Three Men on a Horse]$_G$ is a [play]$_C$ by [George Abbott]$_G$ and [John Cecil Holm]$_E$. . . .<br>***Supporting Document 2:*** [***George Abbott***]$_T$<br>[George Francis Abbott]$_E$ ([June 25, 1887 – January 31, 1995]$_E$) was an [American theater producer]$_C$ and [director]$_C$, [playwright]$_C$, [screenwriter]$_C$, and [film director]$_C$ and [producer]$_C$ whose [career]$_C$ spanned [nine decades]$_E$.<br>. . . |
| STEP$_2$:<br>Phrase graph construction |  [P2] screenwri ter<br>[P2] film director<br>[P2] nine decades<br>John Cecil Holm<br>George Francis Abbott<br>George Abbott<br>Three Men on a Horse<br>[P1] play<br>[P2] playwright<br>[P1] winning horse<br>[P1] race<br>[P1] comedy |
| STEP$_3$:<br>Graph pruning & Question phrase identification |  [P2] screenwri ter<br>[P2] film director<br>[P2] nine decades<br>John Cecil Holm<br>play<br>George Francis Abbott<br>George Abbott<br>Three Men on a Horse<br>[P1] play<br>[P2] playwright<br>playwright<br>[P1] winning horse<br>[P1] race<br>[P1] comedy |
| STEP$_4$:<br>Steiner tree computation |  [P2] playwright<br>'George Abbott', 0<br>George Francis Abbott<br>'George Abbott', title, 0<br>George Abbott<br>Three Men on a Horse: 0<br>Three Men on a Horse<br>Three Men on a Horse: 0<br>[P1] play<br>playwright<br>play |

Table 1: A walk-through example of our method, for the question: *"Three Men on a Horse is a play by a playwright born in which year?"* First, STEP extracts noun phrases and uses the subscript (G as grounding, E as named entity, C as noun chunk, T as tile) to indicate the source of the phrases. Next, construct the graph $G$ using co-occurrence and coreference relations between noun phrases extracted from the context. Next, STEP prunes the graph $G$ by only keeping graph components that exactly or fuzzily match with at least one question phrase. We also add edges such as '*[$P_1$] play*' - '*play*' because the node '*[$P_1$] play*' matches with the question noun phrase *'play'* when removing the prefix *[$P_1$]*. The resulting graph is called $RG$. Question phrases in $RG$ are the purple nodes. Lastly, STEP runs the Steiner tree algorithm over $RG$ to identify the Steiner point *'George Abbott'* and *'George Francis Abbott'*, the nodes highlighted in blue, which are the correct bridge phrases. Due to limited space, we only show the source of each edge in the computed Steiner tree listed in the bottom row.

there is higher chance for the second hop evidence *"George Francis Abbott (June 25, 1887 – January 31, 1995) was an American theater producer and director, playwright, screenwriter. . . "* to be retrieved, while it is less likely to be ranked at the top using the original question as the query.

The expanded query assists a retrieval model to bring more and most relevant set of sentences to the top. The top ranked sentences are then fed into the downstream reader model for answer prediction.

Our method is agnostic to the retrieval algorithm used. For the purpose of evaluating the effectiveness of utilizing STEP as a query expansion method, we choose two off-the-self retrieval models. One is a widely used traditional information retrieval (IR) model (BM25 (Robertson and Zaragoza, 2009)); the second is a transformer-based neural dense retrieval model (cross-encoder (Reimers and Gurevych, 2019)). Specifically, we use a cross-encoder model pre–trained with passages from the MS MARCO dataset (Nguyen et al., 2016), a different dataset from HotpotQA. MS MARCO is a large scale IR corpus, which has become a common starting point for building transformer-based ranking models before further fine-tuning on in-domain and task-specific data (Yates et al., 2021).

## 3.5. Answer Prediction

To further evaluate the impact of STEP on retrieving relevant evidences, we feed the top ranked set of sentences into a reader model for answer extraction. We use the Longformer model as the reader, since it is one of the open-sourced models at the top of the HotpotQA leaderboard. Following (Beltagy et al., 2020), we fine-tune the Longformer model with HotpotQA's training data by concatenating the input query and context sentences in the following format:

$[CLS][Q]$ Query $[/Q][SEP]$ $[T]$title$_1$ $[/T]$ $sent_{11}$ $[/S]$ $sent_{12}[/S]\ldots[SEP]$ $[T]$title$_2$ $[/T]sent_{21}[/S]sent_{22}[/S]\ldots$

where $[Q]$ and $[/Q]$ mark the start and end of the query, $[T]$ and $[/T]$ mark the start and end of the title of the current document, and $[/S]$ marks the end of a sentence. For answer prediction, a classification layer is applied over the $[CLS]$ token for question type classification, and a linear transformation is applied to each token for the prediction of start and end of the answer span.

## 3.6. Post-hoc Reasoning

STEP can also serve as a post-hoc explanation module, for answers provided by other QA components. Post-hoc bridge phrase identification follows the same processes described above, the only difference being that we handle the provided answer similarly to the question. That is, we also ground to answer text during noun phrases identification, we maintain graph components containing nodes matching with at least a question phrase or the answer, and, lastly, we also add the answer node to $RG$ if it does not already exist. Then the Steiner tree algorithm is applied to find the minimal sub-graph that connects all question phrase nodes and the answer node, in which the post-hoc bridge phrases are identified. With the identified bridge phrases, we generate the top ranked evidences by cross-encoder as sentence-level explanation, using a similar query expansion technique. The differences are: first, we replace the *wh*-words in the question with the known answer; otherwise, we append the answer to the question. Next, we append the post-hoc bridge phrases to the query.

# 4. Experiments and Results

**Dataset:** To validate the proposed method, we ran a series of experiments using the HotpotQA dataset (Yang et al., 2018). HotpotQA contains multi-hop questions created by human annotators using documents from Wikipedia as the information sources. Importantly, the questions are designed to only be answerable by combining information from two documents, and require to bridge documents via a concept or entity mentioned in both documents.

| **Bridge Questions** |
|---|
| **Question**: Bordan Tkachuk was the *CEO* of a company that provides what sort of products?<br>**Answer**: IT products and services |
| **Evidences:**<br>1. ***Bordan Tkachuk***: Bordan Tkachuk is a British business executive, the former *CEO* of Viglen, also known from his appearances on the BBC-produced British version of "The Apprentice," interviewing for his boss Lord Sugar.<br>2.***Viglen***: Viglen Ltd provides IT products and services, including storage systems, servers, workstations and data/voice communications equipment and services. |
| **Comparison Questions** |
| **Question:** Which American singer and songwriter has *a mezzo-soprano vocal range*, Tim Armstrong or Tori Amos?<br>**Answer**: Tori Amos |
| **Evidences:**<br>1.***Tim Armstrong***: He is best known as the singer / guitarist for the punk rock band Rancid and hip hop/punk rock supergroup the Transplants.<br>2.***Tori Amos***: Tori Amos (born Myra Ellen Amos, August 22, 1963 ) is an American singer-songwriter, pianist and composer. She is a classically trained musician with *a mezzo-soprano vocal range*. |

Table 2: Examples of bridge and comparison questions from HotpotQA. Bridge phrases are colored in blue (for the bridge question).

HotpotQA contains two question categories: *bridge-type questions*, in which an intermediate entity (i.e., the bridge phrase) is needed to be retrieved before inferring the answer; and *comparison-type questions*, which compare two provided entities. Table 2 shows examples of each of the two question types, with the bridge phrases colored in blue (where applicable).

Given the focus of the proposed work, we use solely the bridge questions in our evaluation.[6] We conduct the evaluation of our unsupervised method STEP on the 5918 bridge-type questions out of the 7,405 examples from the development partition of HotpotQA (Yang et al., 2018) dataset in the distractor setting.

Each question in HotpotQA is supported by two documents, and provided with ground-truth supporting sentences, which enables us to evaluate our approach for both evidence retrieval and the actual QA task.

[6] On average, comparison questions are easier to answer because the necessary information (i.e., the two entities to be compared) are present in the question.

**Experiments:** We demonstrate that STEP identifies bridge phrases that help catch more relevant information for answering multi-hop questions and providing post-hoc reasoning explanations with the following experiments:

**Experiment 1 (evidence retrieval):** In this experiment, we evaluate our method as query expansion for evidence retrieval, i.e., we expand the original question with the bridge phrase(s) identified by STEP. We couple our query expansion strategy with both traditional and neural information retrieval algorithms for evidence retrieval.

**Experiment 2 (question answering):** We use the outputs of the above evidence retrieval components as context for QA, and evaluate the impact of this improved context on answer extraction.

**Experiment 3 (bridge phrases):** To account for the possibility that our bridge phrases yield better evidence sentences and/or answers by mistake, we manually evaluate the bridge phrases generated by our method on a sample of the questions.

**Experiment 4 (explanations):** We evaluate the capacity of our method to provide post-hoc explanations for the situations when an answer exists or is provided by another method. In this experiment, we manually evaluate the quality of the explanations provided by our method for a sample of questions using the gold answers from the dataset.

### 4.1. Experiment 1: Evidence Retrieval

Note that STEP is agnostic to the downstream evidence retrieval component. STEP serves as a query expansion component, where the original HotpotQA question is expanded with the bridge phrases proposed by STEP. We test this query expansion with both a traditional information retrieval model (BM25[7] (Trotman et al., 2014)) and a transformer-based neural retrieval model (cross-encoder[8] (Reimers and Gurevych, 2019)).

Table 3 lists evidence retrieval performance for both strategies when retrieving $k\epsilon \{2, 3, 5, 10, 20\}$ sentences on the bridge questions from the development partition of the HotpotQA dataset. The same results are summarized in Figure 3.

These results highlight several observations. First, the neural retriever performs consistently better than BM25. This is not a surprise: these multi-hop questions exhibit a large "lexical chasm" (Berger et al., 2000), which is better bridged by neural methods. Second, for both traditional and neural retrieval, STEP improves evidence retrieval performance in all settings. This demonstrates that STEP is capable to retrieve additional information and further bridge the information gap that is not modeled by neural retrievers.

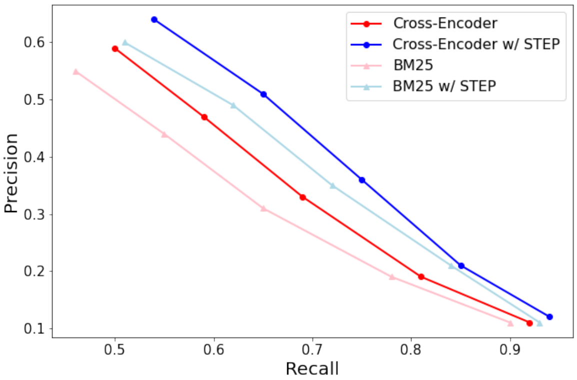


Figure 3: Precision-recall curve for evidence retrieval, when STEP is coupled with a traditional information retrieval component (BM25), or with a neural one (cross-encoder).

### 4.2. Experiment 2: Question Answering

In this experiment, we provide the question along with the retrieved $k\epsilon \{5, 10, 20\}$ sentences by the cross encoder with STEP from the previous step as context to the Longformer model (Beltagy et al., 2020) for answer extraction. We compare the question answering performance against using context obtained from multiple baseline and ceiling strategies.

The two baseline strategies used are:

**Random:** This baseline use a set of $k$ sentences randomly selected from the HotpotQA documents associated with the corresponding question.

**Question-only:** This strong baseline relies on the neural retriever to retrieve evidence sentences using solely the original HotpotQA question as query.

The two ceiling strategies used are:

**SF only:** This strategy uses the ground-truth supporting sentences.

**Oracle:** This ceiling strategy uses queries expanded with oracle bridge phrases extracted directly from the ground-truth supporting sentences, based on several heuristics that identify bridge phrases as: (1) a phrase shared by two supporting sentences; (2) a phrase extracted from a supporting sentence that is also a title phrase of another supporting document; (3) a phrase extracted from a supporting sentence that has an inclusive relationship with a phrase from another supporting sentence.

Table 4 lists overall QA results for the bridge-type questions in the development partition of HotpotQA. These results indicate that the Longformer using context sentences retrieved with STEP expanded query consistently obtains better QA performance than the strong baseline that uses sentences retrieved by the cross-encoder retriever using just the original question (the "Question-only" configurations). Further, our best configuration (cross-encoder with STEP) outperforms the Random baseline considerably, and approaches the

[7] https://pypi.org/project/rank-bm25/

[8] https://huggingface.co/cross-encoder/ms-marco-MiniLM-L-6-v2

|  | Prec@2 | Prec@3 | Avg Prec | Recall@2 | Recall@3 | Recall@5 | Recall@10 | Recall@20 |
|---|---|---|---|---|---|---|---|---|
| Cross Encoder | 0.59 | 0.47 | 0.64 | 0.50 | 0.59 | 0.69 | 0.81 | 0.92 |
| Cross Encoder w/ STEP | **0.64** | **0.51** | **0.69** | **0.54** | **0.65** | **0.75** | **0.85** | **0.94** |
| BM25 | 0.55 | 0.44 | 0.60 | 0.46 | 0.55 | 0.65 | 0.78 | 0.90 |
| BM25 w/ STEP | 0.60 | 0.49 | 0.66 | 0.51 | 0.62 | 0.72 | 0.84 | 0.93 |

Table 3: Evidence retrieval performance (precision@$k$ and recall@$k$) for all bridge questions in the development partition of HotpotQA, when STEP is coupled with a traditional IR component (BM25), or with a neural one (cross-encoder).

|  | EM | F1 | P | R |
|---|---|---|---|---|
| Random 5 | 0.11 | 0.18 | 0.19 | 0.19 |
| Random 10 | 0.19 | 0.29 | 0.3 | 0.29 |
| Random 20 | 0.3 | 0.44 | 0.46 | 0.45 |
| Question-only Top 5 | 0.33 | 0.48 | 0.5 | 0.49 |
| Question-only Top 10 | 0.41 | 0.57 | 0.6 | 0.58 |
| Question-only Top 20 | 0.49 | 0.67 | 0.7 | 0.69 |
| SF only | **0.56** | **0.77** | **0.79** | **0.79** |
| Oracle Top 5 | 0.49 | 0.67 | 0.7 | 0.68 |
| Oracle Top 10 | 0.52 | 0.71 | 0.74 | 0.72 |
| Oracle Top 20 | 0.54 | 0.73 | 0.76 | 0.75 |
| STEP Top 5 | 0.4 | 0.55 | 0.58 | 0.57 |
| STEP Top 10 | 0.45 | 0.62 | 0.65 | 0.64 |
| STEP Top 20 | **0.51** | **0.69** | **0.72** | **0.71** |

Table 4: Exact match, F1, precision, and recall scores for QA performance using Longformer for answer extraction, over contexts retrieved with various strategies.

performance of the "Oracle" setting, i.e., when bridge phrases are extracted directly from the correct supporting sentences. This further suggests that STEP identifies new and useful information that is missed by transformer networks.

## 4.3. Experiment 3: Bridge Phrases

The above experiments demonstrated that STEP identifies expansion terms that augment the question to help in increasing the relevance of evidences that have low lexical or semantic overlap with the initial question in general. To take a closer look at the quality of the bridge phrases proposed by STEP, and to investigate whether they do assist to connect the information gap between the questions and their answers, we randomly selected 100 questions from the dataset, and asked two human annotators to annotate the extracted bridge phrases as: correct (if they bridge the necessary connection between question and answer), incorrect (if they do not), and partially correct (if only some of correct bridge phrases are identified). According to the human annotators, the average accuracy of the bridge phrases generated by STEP is 76.3%. The Kappa inter-annotator agreement was 46%, which is ranked as moderate agreement. We consider this agreement respectable given the complexity and the ambiguity of the task (i.e., there may be multiple ways to answer a given question). Table 5 lists a few examples of bridge phrases extracted by STEP.

We manually inspected examples where the two annotators disagreed, and found that disagreement happens in the following cases: (a) lexical ambiguity or overlap. As an example, the correct bridge phrase for the question (3) in Table 5 is *'Boston Lincolnshire'*, but STEP extracts *'Boston'*, which lexically overlaps with the correct phrase; (b) there is more than one bridge phrase. For example, question (4) in Table 5 has two bridge phrases: *'Old Frisian'* (i.e., *'the Western Germanic language'*) and *'Kloster Muhde'* (i.e., *'the small settlement'*). While *'Old Frisian'*, the major bridge phrase that connects all the question phrases, has been identified correctly, one annotator marked as correct, and the other marked as partially correct for missing *'Kloster Muhde'*.

We also inspected the erroneous cases, and found two common causes of error: (a) STEP found the answer instead of the bridge phrase, as in question (6) in Table 5, while the correct bridge phrase is *'Catuvellauni'*; and (b) STEP does not identify a bridge phrase because all the mentioned phrases in the question are well-connected. For example, for question (5) in Table 5, *'On My Mind'* is the correct bridge phrase (i.e., *the song in Ellie Goulding's third studio album that was written by her and Max Martin, Savan Kotecha and Ilya Salmanzadeh*). However, the question phrases *'third studio album', 'writers', 'Delirium'* are all connected to *'Ellie Goulding'* by co-occurrence. There is no need of a Steiner point to bridge them.

Overall, this manual evaluation showed that STEP is capable to identify high quality bridge phrases that connect the information gap between the question and the relevant context for most questions.

## 4.4. Experiment 4: Post-hoc Explanations

Lastly, we evaluate STEP's capacity to provide auxiliary *post-hoc* explanations to interpret answers provided by an external component (be it human or machine). Table 6 show an example of post-hoc explanation generated by coupling STEP with the cross-encoder reranker. Given the answer *'Murray Hill'*, one will fast locate the candidate evidence #1, and find the underlined bridge phrase *'Bell Labs'*, and then gap the reasoning by locating another evidence that would confirm *'Bell Labs'* is *"the American research and scien-*

|  | Question | Answer | Bridge Phrases (STEP) | Annotations |
|---|---|---|---|---|
| (1) | Ralph Hefferline was a psychology professor at a university that is located in what city? | New York City | Columbia University | (Correct, Correct) |
| (2) | The Vermont Catamounts men's soccer team currently competes in a conference that was formerly known as what from 1988 to 1996? | the North Atlantic Conference | America East Conference | (Correct, Correct) |
| (3) | According to the 2001 census, what was the population of the city in which Kirton End is located? | 35,124 | Boston | (Correct, Partial) |
| (4) | When was the Western Germanic language spoken from which the small settlement situated on the river Leda opposite Leer derives its name? | between the 8th and 16th centuries | Old Frisian<br>English language | (Correct, Partial) |
| (5) | Ellie Goulding worked with what other writers on her third studio album, Delirium? | Max Martin, Savan Kotecha and Ilya Salmanzadeh | – | (Incorrect, Incorrect) |
| (6) | This Celtic ruler who was born in AD 43 ruled southeastern Britain prior to conquest by which empire? | Roman | Roman conquest of Britain | (Correct, Incorrect) |

Table 5: Examples of bridge phrases STEP identified, and human evaluation from two annotators. Due to limited space, supporting facts the annotators used to evaluate the bridge phrases are not listed here.

*tific development company where Ravi Sethi worked as computer scientist"*. This is done efficiently by looking for the candidate evidence that connects *'Bell Labs'* with one of the rest question phrases *'the American research and scientific development company'*, *'Ravi Sethi'*, *'computer scientist'*, and thus locates candidate evidence #4 and #2 immediately, which closes the reasoning loop. In this example, one would skip the candidate evidence #3 because it does not contain any connection between the noun phrases of interested.

For this evaluation, we provided explanations as in Table 6 with top 10 candidate evidences for 50 random sampled questions, and asked the annotators to evaluate the quality of the generated explanations. The annotations report 44.5 out of the 50 questions were provided with high quality of explanations, and STEP identified the correct post-hoc bridge phrases for 48 questions. Considering the reranker we used is a model trained in a zero-short manner, it is likely that an even higher quality of explanation would be generated when using a more powerful ranker trained in-domain.

## 5. Conclusion

We proposed an unsupervised approach for the identification of bridge phrases in multi-hop question answering. Our method constructs a graph of noun phrases from the question and the available context, and applies the Steiner tree algorithm to identify the minimal subgraph that connects all question phrases. We extract as bridge phrases nodes in this graph that are not any of the question phrases. Our method can be coupled with any downstream QA component, i.e., it can be used as query expansion for evidence retrieval; it can be used to generate enhanced context for answer prediction; and it can be used to generate post-hoc explanations for given answers. Using the HotpotQA dataset, we demonstrate that our method yields improved results in all these scenarios, for multiple types of downstream components.

**Question**: In which city are the headquarters of the American research and scientific development company where Ravi Sethi worked as computer scientist located?

**Answer**: Murray Hill

**Top ranked evidence candidates:**

1. ***Bell Labs***: Its headquarters are located in Murray Hill, New Jersey, in addition to other laboratories around the rest of the United States and in other countries.

2.***Ravi Sethi***: Ravi Sethi (born 1947) is an Indian computer scientist retired from Bell Labs and president of Avaya Labs Research.

3. ***Ravi Sethi***: He also serves as a member of the National Science Foundation's Computer and Information Science and Engineering (CISE) Advisory Committee.

4. ***Bell Labs***: Nokia Bell Labs (formerly named AT&T Bell Laboratories, Bell Telephone Laboratories and Bell Labs) is an American research and scientific development company, owned by Finnish company Nokia.

5. ***Ravi Sethi***: He is best known as one of three authors of the classic computer science textbook "", also known as the "Dragon Book".

Table 6: Examples of post-hoc explanation. Phrases in the Steiner tree computed by STEP are underlined, from which bridge phrases that do not appear in question nor answer are marked in blue. Candidate evidences ranked at positions 1, 2, and 4 are the correct supporting facts.

## 6. Acknowledgements

This work was partially supported by the Defense Advanced Research Projects Agency (DARPA) under the World Modelers program, grant number W911NF1810014. Mihai Surdeanu declares a financial interest in lum.ai. This interest has been properly disclosed to the University of Arizona Institutional Review Committee and is managed in accordance with its conflict of interest policies.

## 7. Bibliographical References

Alromima, W., Moawad, I. F., Elgohary, R., and Aref, M. (2016). Ontology-based query expansion for arabic text retrieval. *Int. J. Adv. Comput. Sci. Appl*, 7(8):223–230.

Asai, A., Hashimoto, K., Hajishirzi, H., Socher, R., and Xiong, C. (2020). Learning to retrieve reasoning paths over wikipedia graph for question answering.

Azad, H. K. and Deepak, A. (2019). Query expansion techniques for information retrieval: a survey. *Information Processing & Management*, 56(5):1698–1735.

Beltagy, I., Peters, M. E., and Cohan, A. (2020). Longformer: The long-document transformer.

Berger, A., Caruana, R., Cohn, D., Freitag, D., and Mittal, V. (2000). Bridging the lexical chasm: statistical approaches to answer-finding. In *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 192–199.

Bouadjenek, M. R., Hacid, H., Bouzeghoub, M., and Vakali, A. (2016). Persador: personalized social document representation for improving web search. *Information Sciences*, 369:614–633.

Cao, N. D., Aziz, W., and Titov, I. (2019). Question answering by reasoning across documents with graph convolutional networks.

Chen, D., Fisch, A., Weston, J., and Bordes, A. (2017). Reading wikipedia to answer open-domain questions.

Das, R., Dhuliawala, S., Zaheer, M., and McCallum, A. (2019). Multi-step retriever-reader interaction for scalable open-domain question answering.

Ding, M., Zhou, C., Chen, Q., Yang, H., and Tang, J. (2019). Cognitive graph for multi-hop reading comprehension at scale. *arXiv preprint arXiv:1905.05460*.

Esposito, M., Damiano, E., Minutolo, A., De Pietro, G., and Fujita, H. (2020). Hybrid query expansion using lexical resources and word embeddings for sentence retrieval in question answering. *Information Sciences*, 514:88–105.

Fang, Y., Sun, S., Gan, Z., Pillai, R., Wang, S., and Liu, J. (2019). Hierarchical graph network for multi-hop question answering. *arXiv preprint arXiv:1911.03631*.

Feldman, Y. and El-Yaniv, R. (2019). Multi-hop paragraph retrieval for open-domain question answering.

Guo, L., Su, X., Zhang, L., Huang, G., Gao, X., and Ding, Z. (2018). Query expansion based on semantic related network. In *Pacific Rim International Conference on Artificial Intelligence*, pages 19–28. Springer.

Hartmanis, J. (1982). Computers and intractability: a guide to the theory of np-completeness (michael r. garey and david s. johnson). *Siam Review*, 24(1):90.

Honnibal, M., Montani, I., Van Landeghem, S., and Boyd, A. (2020). spaCy: Industrial-strength Natural Language Processing in Python.

Jiang, Y. and Bansal, M. (2019). Self-assembling modular networks for interpretable multi-hop reasoning. *arXiv preprint arXiv:1909.05803*.

Min, S., Zhong, V., Zettlemoyer, L., and Hajishirzi, H. (2019). Multi-hop reading comprehension through question decomposition and rescoring.

Nakade, V., Musaev, A., and Atkison, T. (2018). Preliminary research on thesaurus-based query expansion for twitter data extraction. In *Proceedings of the ACMSE 2018 Conference*, pages 1–4.

Nguyen, T., Rosenberg, M., Song, X., Gao, J., Tiwary, S., Majumder, R., and Deng, L. (2016). Ms marco: A human generated machine reading comprehension dataset. In *CoCo@ NIPS*.

Qi, P., Lin, X., Mehr, L., Wang, Z., and Manning, C. D. (2019). Answering complex open-domain questions through iterative query generation. *arXiv preprint arXiv:1910.07000*.

Reimers, N. and Gurevych, I. (2019). Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.

Robertson, S. and Zaragoza, H. (2009). *The probabilistic relevance framework: BM25 and beyond*. Now Publishers Inc.

Takahashi, H. et al. (1980). An approximate solution for the steiner problem in graphs.

Trotman, A., Puurula, A., and Burgess, B. (2014). Improvements to bm25 and language models examined. In *Proceedings of the 2014 Australasian Document Computing Symposium*, pages 58–65.

Tu, M., Wang, G., Huang, J., Tang, Y., He, X., and Zhou, B. (2019). Multi-hop reading comprehension across multiple documents by reasoning over heterogeneous graphs.

Wang, H., Zhang, Q., and Yuan, J. (2017). Semantically enhanced medical information retrieval system: a tensor factorization based approach. *Ieee Access*, 5:7584–7593.

Wu, H., Wu, W., Zhou, M., Chen, E., Duan, L., and Shum, H.-Y. (2014). Improving search relevance for short queries in community question answering. In *Proceedings of the 7th ACM international conference on Web search and data mining*, pages 43–52.

Xiao, Y., Qu, Y., Qiu, L., Zhou, H., Li, L., Zhang, W., and Yu, Y. (2019). Dynamically fused graph network for multi-hop reasoning.

Yang, Z., Qi, P., Zhang, S., Bengio, Y., Cohen, W. W., Salakhutdinov, R., and Manning, C. D. (2018). Hotpotqa: A dataset for diverse, explainable multi-hop question answering.

Yates, A., Nogueira, R., and Lin, J. (2021). Pretrained transformers for text ranking: Bert and beyond. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*, pages 1154–1156.